NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ASSESSING THE POSSIBLE RETURN ON INVESTMENT
RESULTING FROM UPGRADING A SUBSYSTEM

by

Chang Tun-Jen

March 1993

Thesis Advisors: Donald P. Gaver
Patricia A. Jacobs

Approved for public release; distribution is unlimited.

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
March 1993

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Assessing the Possible Return on Investment Resulting from Upgrading a Subsystem

5. FUNDING NUMBERS

6. AUTHOR(S)
CHANG, Tun-Jen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This thesis develops a decision aid to assist in assessing the cost effectiveness of upgrading a subsystem. The procedures developed in this thesis are to estimate the time of onset and the magnitude of the degradation of a subsystem and to estimate the best time to upgrade the subsystem. Two procedures are considered to estimate the time of onset of subsystem degradations and the magnitude of the degradation. One is maximum likelihood; the other is a Bayesian procedure. These estimates are then used in a cost model to estimate the cost of remaining with the current subsystem for the remaining planned lifetime of the system. A comparison of this cost with that of investing in the upgraded subsystem can be used to obtain a best time to invest in the upgraded subsystem. Procedures to assess the uncertainty of the cost advantage of upgrading the subsystem are also studied to give further information to the decision maker.

14. SUBJECT TERMS
Decision Aid, Upgrading a subsystem

15. NUMBER OF PAGES
126

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std 239-18

UNCLASSIFIED

Approved for public release; distribution is unlimited.

ASSESSING THE POSSIBLE RETURN ON INVESTMENT RESULTING FROM UPGRADING A SUBSYSTEM

by

Chang Tun-Jen
Lieutenant Commander, Chinese Navy
B.S., Chung Cheng Institute of Technology, 1982

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1993

Author: CHANG TUN-JEN
Chang Tun-Jen

Approved by: Donald P. Gaver
Donald P. Gaver, Thesis Advisor

Patricia A. Jacobs
Patricia A. Jacobs, Thesis Advisor

So Young Sohn
So Young Sohn, Second Reader

P. Purdue
Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

This thesis develops a decision aid to assist in assessing the cost effectiveness of upgrading a subsystem. The procedures developed in this thesis are to estimate the time of onset and the magnitude of the degradation of a subsystem and to estimate the best time to upgrade the subsystem. Two procedures are considered to estimate the time of onset of subsystem degradation and the magnitude of the degradation. One is maximum likelihood; the other is a Bayesian procedure. These estimates are then used in a cost model to estimate the cost of remaining with the current subsystem for the remaining planned lifetime of the system. A comparison of this cost with that of investing in the upgraded subsystem can be used to obtain a best time to invest in the upgraded subsystem. Procedures to assess the uncertainty of the cost advantage of upgrading the subsystem are also studied to give further information to the decision maker.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

A subsystem of an aircraft total system, for example an APS-80 antenna subsystem within the P-3B aircraft system, may tend to exhibit unfavorable reliability or maintenance cost properties, beginning at some random point in time. The evidence of such degradation of a subsystem suggests the possible economic and operational value of subsystem upgrade. The decision to upgrade a subsystem will, at least partially, be based on a comparison of the costs of remaining with the current subsystem over its remaining time horizon, and those of investing in the upgraded subsystem for use in the remaining time horizon.

The cost of remaining with the current subsystem must be estimated using available data. It will depend on the estimate of the time of onset of subsystem degradation and the estimate of the magnitude and evolution of the degradation over time. The costs of investing in the upgraded subsystem are obtained from another source. A comparison of the cost (1) of remaining with the current subsystem with (2) investing in the upgraded subsystem can be used to obtain a "best" time to invest in the upgraded subsystem. This time can be beyond the system's

remaining planned life, in which case the upgrade is unadvisable.

B. CURRENT METHOD AND MODEL

The economic analysis program (ROI) in the Automated Management Indicator System (AMIS) is designed to compare an existing (current) subsystem with an improved (upgraded) subsystem. The program compares the projected cumulative cost of the existing subsystem to the projected cost of the improved subsystem. After computing these costs, the program computes the number of months until the cumulative cost of the existing subsystem is equal to the cumulative cost of the improved subsystem when the investment cost and upgrade schedule are considered. If this break-even time exists, the time is printed on the screen. If this break-even point occurs before the end of the planned system life then the gains from upgrade are potentially available; if not then the upgrade is not likely to be worthwhile.

Most of the input data for the economic analysis program comes from the NALDA data base. While this data base collects the measures of subsystem performance over a long period of time, the ROI program only uses an average of the data for the last 24 months to estimate the predicted future subsystem performance. It uses this average to compute the future cost of the current subsystem.

Two features of the ROI model are considered in the development of a new model. One feature is that average values of measures of subsystem performance may be correct for local cost estimation, but may not represent the evolution of the subsystem degradation. The other feature is that the ROI program only computes the required time to cover the upgrade investment if the decision is made to upgrade the subsystem immediately. It may be better to wait to initiate subsystem upgrade in order to make more certain that an adverse trend exists. The model investigated in this thesis gives a best time to initiate subsystem upgrade based on estimated costs.

C. NEW APPROACH

The approach of this thesis is to consider the time series of a measure of subsystem performance and to estimate the time of onset of subsystem degradation and the magnitude of the degradation as time advances. These estimates are then used in a cost model to estimate the cost of remaining with the current subsystem for the remaining planned lifetime of the subsystem. We call this planned lifetime the time horizon.

Two procedures are considered to estimate the time of onset of subsystem degradation and the magnitude of the degradation. Both are based on a simple change-point model that assumes that the degradation may begin at some time point and increase linearly thereafter. The change-point and the rate of degradation must be estimated from data; two

procedures are used. One is maximum likelihood; the other is a Bayesian procedure. Details of these procedures appear in Appendices B and C. A cost model is formulated to be compatible with that used in the AMIS ROI program. A description of the cost model appears in Appendix D. Procedures to assess the variability of the cost advantage of upgrading the subsystem are studied. The procedures used are described in the Appendix E.

A decision aid using the methodologies is programmed in TURBO PASCAL. The resulting program is called UPGRADE.PAS. The program's source code and user documentation are attached in Appendix H. The program can simulate data and provide graphical output.

The methodologies are used to analyze simulated data and data from a radar transmitter on the F-14A. The data were supplied by C. Wrestler of NAVAIR (419) and come from the NALDA data base. These data appear in Appendix F. Results from the analysis appear in Appendix G.

A brief discussion of the models and procedures appears in the next chapter. Information concerning the input data for this program appears in Chapter III. The results of analyses conducted using the PASCAL program will be discussed in Chapter IV.

II. MODELS AND METHODS

The model consists of two parts. One is a statistical model and the other is a cost model. The type of statistical model has been recognized early and studied by many under the name of *changepoint problem* (Carlin, Gelfand and Smith 1992). The combination of the changepoint problem with a cost model to estimate future costs to make a decision is novel. In this chapter we describe the statistical model and the cost model.

A. STATISTICAL MODEL

We consider a model for a measure of subsystem performance in successive periods of time. Two such measures of performance are the mean number of failures in a time period or the mean number of maintenance actions in a time period.

Consider a sequence of random variables with the following structure:

1. $X_1, X_2, \ldots, X_C$ are identically and independently distributed, while
2. $X_{C+1}, X_{C+2}, \ldots, X_t$ exhibit a linear trend.

The time of onset of subsystem degradation, C, called the changepoint, will realistically be unknown, as will the magnitude of the linear trend. We will assume

$$X_i \sim N(\mu,\sigma^2), \qquad 0 \le i \le C;$$
$$\sim N(\mu+(i-C)\eta,\sigma^2), \quad C+1 \le i. \qquad (2.1)$$

This is shorthand for the assumption that X_i is normally/Gaussianly distributed with mean μ and variance σ^2 up to the changepoint time C and is normally distributed thereafter, but with mean that grows (with the slope, η) linearly thereafter; μ is the mean number of failures (or maintenance actions) in each time period before the onset of degradation; C is the time of onset of degradation; η is the slope of the linear trend after degradation; and the variance σ^2 is a measure of the variability of the actual number of failures (or maintenance actions) about the true mean. This model should be appropriate for subsystems whose mean failure/maintenance rate per time period, e.g., month, is reasonably large, but whose variance is relatively unchanged when and if a change in the mean occurs.

Figure 1 shows data simulated from the above model with mean μ=4, slope η=1.5, variance σ^2=1, and changepoint C=10. The normal random variables are generated using the Box-Muller technique (see G.S. Fishman 1978) which is described in Appendix A.

In Figure 1 the x's represent the actual data (e.g., mean number of failures or mean number of maintenance actions in a month), while the dashed lines represent the true, but hidden trend.

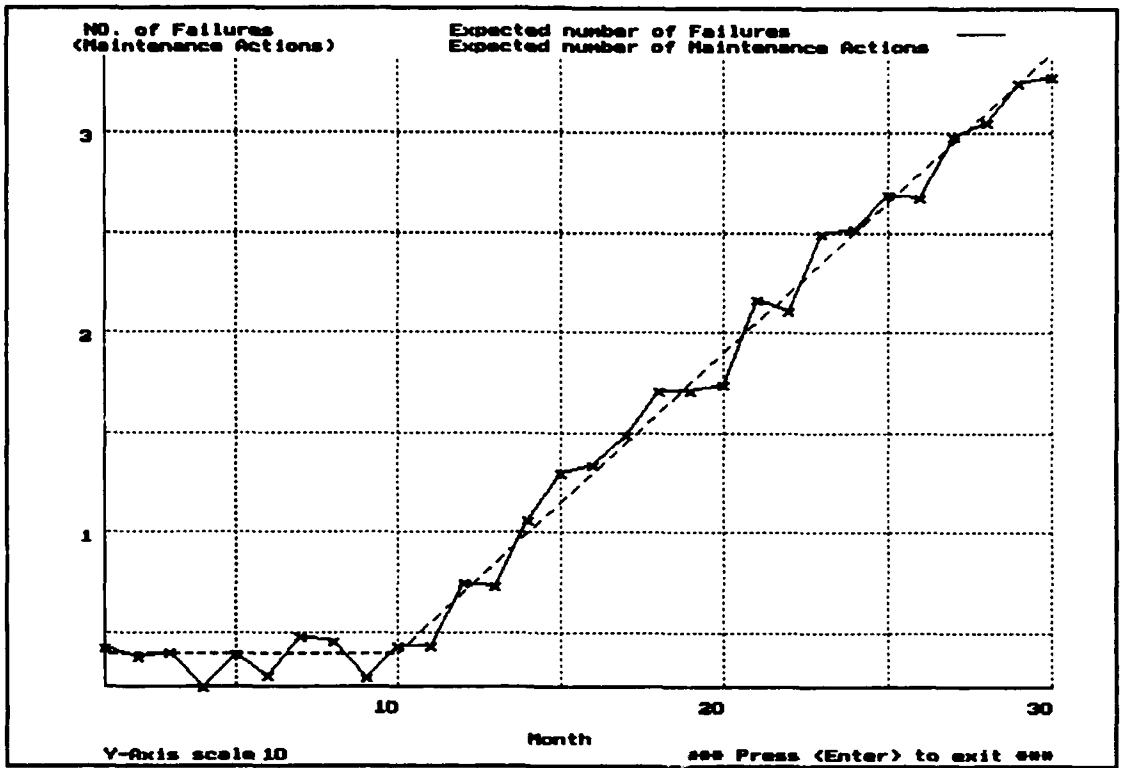

Figure 1. Simulated data (μ=4,η=1.5,σ^2=1,C=10).

It is plain that there is little evidence of any change in the demand level until time t=12 at the earliest, when a retrospective look suggests that a change took place at time t=10. Successively more confirmation is given by observations after t=14. If the trend continues as suggested, greater and greater confirmation of its direction and magnitude becomes available; this can be quantified by the statistical methods described in Appendix B (maximum likelihood method) or Appendix C (Bayesian method). The statistical methods provide estimates of the true trend (denoted by the dashed line above) based on the mean number of failures or mean number of maintenance actions observed (the x's above); the estimates

are of the time of onset of subsystem degradation and the magnitude of that degradation.

B. COST MODEL

The following is a brief discussion of the cost model which will have as input the estimates of the time of onset of subsystem degradation and the magnitude of the trend. The cost model includes a fixed cost and schedule for upgrading the subsystem as well as costs for each failure, maintenance action, and AV-DLR action for both the current and upgraded subsystems. The detailed cost model which depends on these estimates is presented in Section B of Appendix D. The model is briefly described in the following.

There are three basic unit costs incurred by the current subsystem. There is a cost c_{OF} (respectively c_{OM}) incurred each time the subsystem fails, (respectively requires a maintenance action). There is also a cost of c_{OA} for each AV-DLR action. These unit costs are computed from cost data in the ROI program. The details of their calculation appear in Section A of Appendix D.

The proposed upgraded subsystem also has unit costs c_{NF} (respectively c_{NM}) incurred each time the subsystem fails, (respectively requires a maintenance action). There is also a cost of c_{NA} for each AV-DLR action. Details of their calculation also appear in Section A of Appendix D.

An additional cost is an initial fixed cost for the upgrade, c_F. In addition to the above costs, there is a required lead time L to prepare for the upgrade. There is also a period of time J to install the upgrade; this period of time depends on the installation rate. Details of their calculation appear in Section C of Appendix D.

C. ASSESSING UNCERTAINTY

Since the estimates of the true trend have variability, the estimated future cost of the current subsystem will also have variability. As with the estimated future cost of the true trend, one can expect the estimates of the future subsystem cost to be quite variable until sometime after the onset of degradation; this variability is due to uncertainty in the estimates of the true trend.

It is important to consider this uncertainty in the assessment of whether or not to upgrade the current subsystem. For example, it may be that the estimated mean future cost of the current subsystem is larger than that for the upgraded subsystem but that the uncertainty associated with the estimated mean future cost of the current subsystem is high. This may indicate that it is better to wait to accumulate more information concerning the apparent degrading trend before deciding to invest in the upgrade. Procedures to assess the variability of the estimated cost advantage of upgrading the subsystem (future mean cost of the current subsystem minus

future mean cost of the upgraded subsystem) are described in Appendix E.

III. INPUT DATA

The input data for a radar transmitter on the F-14A that are used in the ROI decision aid are attached in Appendix F. The data come from the NALDA data base. The model developed in this thesis and implemented in the FCT.PAS program is designed to use these data to provide a rational time at which to upgrade the subsystem, given current information. The main difference between the ROI decision aid and the procedures developed here is in the forecasting of the future performance of the current subsystem. The ROI procedure uses the average values of performance measures for the last 24 months to forecast the current subsystem's future behavior. The model in this thesis estimates a time of onset of subsystem degradation and the magnitude of the degradation using the time series of the measures of performance.

The two time series considered are obtained as follows. Table 2 in Appendix F gives the total number of aircraft and the total flight hours per month. Because the total number of aircraft and total flight hours per month change over time, we use the mean number of failures instead of the real number of failures in each month to estimate the performance of the subsystem. This value is the mean monthly flight hours, computed as the average number of systems times the average use per month appearing in Table 1, divided by the mean flight

hours between failure (column MFHBF in Table 3) for each month. The average number of systems and the average use per month use the average values of measures for the last 24 months in Table 2. The mean number of maintenance actions each month is estimated as the same mean monthly flight hours as above divided by the mean flight hours between maintenance action (column MFHBMA in Table 3).

The number of AV-DLR actions per month depends on the number of BCM's in each month. Examination of the data suggests that the number of BCM's per month is independent of the number of flight hours in that month. Thus, we assume the number of AV-DLR actions has a constant rate per month. We use the average value of BCM's for the last 24 months to estimate the AV-DLR action rate per month. This value is the same as that used in the ROI decision aid. The detailed description of the parameters used in our decision aid is presented in Appendix D.

IV. RESULTS

Appendix G presents the results of using our decision aid. Section A presents results of the procedures using simulated data. Section B presents results of using the procedures on data for a radar transmitter on the F-14A.

A. RESULTS FOR SIMULATED DATA

We first discuss the results for a set of simulated data. Two time series using data simulated from model (2.1) were used. Data for the mean number of failures in each month was simulated from the model with parameters μ_F=225, η_F=4, σ_F^2=144, and C_F=25. Data for the mean number of maintenance actions was simulated from the model with parameters μ_M=440, η_M=5, σ_M^2=225, and C_M=25. The length of both time series is chosen to be 40 months and the time horizon is to be 150 months. These data are chosen to approximately mimic the real data discussed in the next section, but to have more apparent linear trends. The other parameters concerning cost computation were chosen to be equal to those in the next section which also gives a detailed explanation of the computation of the parameters. Figure 2 in Appendix G displays graphs of the two sets of simulated data along with the true mean as a dashed line. A listing of the simulated data appears below Figure 2.

For each time $6 \le t \le 40$ the maximum likelihood procedure and Bayesian procedure were used to estimate the model parameters using data $x_1, x_2, \ldots, x_t$.

The cost of upgrading at time τ in the future $\hat{C}_N(\tau, t)$ and the cost of never upgrading $\hat{C}_O(t)$ were computed using the expressions (D.8) through (D.13) in Appendix D.

The parameters used in the cost model appear in the menu below the listing of the simulated data in Appendix G. A description of how they are used appears in Section A of Appendix D. The time τ which minimizes $\hat{C}_N(\tau, t)$ is computed. If $\hat{C}_O(t) < \min_\tau \hat{C}_N(\tau, t)$, then the minimizing τ is taken to be the horizon time, taken to be 150 in the example. A positive value of the cost advantage of upgrade, $\hat{C}_O(t) - \hat{C}_N(\tau, t)$, indicates that it is better to switch to the new subsystem; the more positive, the greater the estimated advantage of changing to the new subsystem.

Below the listing of the menu in subsection 2 of Appendix G is a listing of the minimizing τ for each time t for the Bayesian procedure. Also displayed is the mean cost advantage of upgrade for the best policy and its two standard deviation bounds for assessing uncertainty; if the minimum cost policy is never to upgrade, then the cost advantage is $\hat{C}_O(t) - \hat{C}_N(0, t)$. Notice that initially the best policy is to do nothing. However, as time increases, the best policy is to begin the procedure to upgrade immediately. This policy is interspersed with the policy to do nothing until time 26. After time 26,

the best policy is always to begin upgrading immediately. The true best policy would be to do nothing up to time 26 and then to start upgrading immediately. Thus the policy using the estimated costs occasionally gives false alarms, (suggesting that the upgrading start before time 26). This behavior suggests that it is prudent to wait for confirmation of a decision to upgrade before starting the upgrade policy.

Figure 3 presents graphs of the cost advantage of upgrading obtained from the Bayesian assessment of cost variability for minimum cost policies from t=6 to t=40 as described in Section B of Appendix E. The graph displays the mean cost advantage of the minimum cost policy and the mean cost advantage plus and minus two standard deviations; if the minimum cost policy is never to upgrade, then the cost advantage is $\hat{C}_O(t)-\hat{C}_N(0,t)$. The width between two bounds can be interpreted as representing an approximate Bayesian posterior density for the true expected or mean cost advantage, given observations up to time t. The width between the bounds becomes smaller as more data accumulates and the uncertainty of the estimates of the changepoint and the degradation rate, η, is reduced. The width of the bounds provide prospective on the risk of changing soon, or waiting. Apparently the chance of making the wrong decision decreases if the decision maker waits, but also the value of making the more nearly correct decision decreases, for there is less time to the horizon. Recall that the true time of onset of

subsystem degradation occurs at time 25. If one waited until the lower confidence bound of the cost advantage of the minimum cost policy becomes positive, then one would wait until time 34 to make a decision to upgrade the subsystem.

Subsection 3 presents the output for the likelihood procedure using the same simulated data. The best policy is always to upgrade immediately after time 26. Comparing the upgrading policies with those from Bayesian procedure, the only difference occurs at time 25; the Bayesian procedure does not recommend upgrading, while the likelihood procedure recommends upgrading immediately. This is before the time of onset of degradation. Figure 4 presents graphs obtained from a bootstrap assessment of the variability of the cost advantage of the best policies from time 6 to time 60. The variability of the cost advantage of upgrading estimated from the likelihood procedure tends to be larger than that from the Bayesian procedure. This result is due to the likelihood procedure's having more variability in the estimated time of onset of the degradation.

Simulation was also used to investigate the behavior of the estimation procedures for two other changepoint models. In one (a jump model)

$$X_i \sim N(\mu, \sigma^2) \qquad 1 \le i \le C,$$
$$\sim N(\mu+\delta, \sigma^2) \qquad C+1 \le i; \tag{4.1}$$

that is, X_i is normally/Gaussianly distributed with mean μ and variance σ^2 up to the changepoint time C and is normally distributed thereafter with an different constant mean $\mu+\delta$. The other model (jump plus linear trend model)

$$X_i \sim N(\mu, \sigma^2) \qquad 1 \le i \le C,$$
$$\sim N(\mu+\delta+\eta(i-C), \sigma^2) \qquad C+1 \le i; \tag{4.2}$$

that is, X_i is normally/Gaussianly distributed with mean μ and variance σ^2 up to the changepoint time C and is normally distributed thereafter with a mean that has a jump δ and then grows (with the slope η) linearly thereafter.

Simulation studies indicate that the maximum likelihood procedures for these last two models yield estimates that are more sensitive to the local behavior of the data than those for the model presented in Chapter II. This sensitivity to local behavior tends to produce more "false alarms" concerning the presence of degradation. The estimator for δ in both of the above models is also greatly influenced by local behavior in the data leading to a large assessment of variability.

B. RESULTS FOR DATA FROM A RADAR TRANSMITTER ON THE F-14A

Results concerning the analysis of data from a radar transmitter on the F-14A appear in Section B of Appendix G.

The mean numbers of failures each month, and the mean numbers of maintenance actions each month, were calculated as described in Chapter III. A listing of the two time series appears below Figure 5 which presents graphs of the two series. The data show considerable variation and tend to increase. Thus the subsystem appears to be degrading over time. There is no way of knowing what actions, if any, were taken in the event that such a tendency was noted. Using the parameters calculated in Appendix F, the cost model for the example has the following features. The fixed cost per failure for the current subsystem is c_{OF}=1,323.6. The fixed cost per maintenance action is c_{OM}=166.8; and the fixed cost per AV-DLR action is c_{OA}=1,120.3. The mean number of AV-DLR actions per month is 7.8. There is a fixed initial cost c_F=5,400,000 for upgrading the subsystem. All costs are in dollars. The upgraded subsystem is assumed to have a mean number of failures of 125 per month, and a mean number of maintenance actions of 485 per month. The fixed cost per failure for the upgraded subsystem is assumed to be c_{NF}=1,323.6. The fixed cost per maintenance action for the upgraded subsystem is c_{NM}=204.4; and the fixed cost per AV-DLR action for the upgraded subsystem, c_{NA}, is the same as current subsystem. The procedures used to compute the cost parameters appearing in this section are described in Appendices D and F. The data were collected over 5 years (January 1987 through December 1991). There is a time horizon, H=180 months, during which

this subsystem or its upgrade will be used, and a lead time, L=60 months, to prepare the upgrade. The decision to upgrade depends on the estimates of the time of onset of subsystem degradation and of the magnitude of the trend. The assessment of the cost of upgrading should reflect the uncertainty of these estimates.

For each time $t \geq 6$, the following policies are considered: upgrade the subsystem at each future time until the time horizon; all potential upgrading times from the present time until the time horizon $H-L$ are considered; that is, if the current time is t=40 then the policies that would upgrade the subsystem at time 40, time 41,..., time 120, (which is $H-L$), are considered. For each current time t, the (estimated) costs of these policies are compared to the (estimated) costs of never upgrading the subsystem. The "optimal" (maximum estimated cost advantage) policy can then be found.

For each time $t \geq 6$, the model considers the data accumulated up to time t and using the data as of that time estimates the time of onset of subsystem degradation and magnitude of the trend. The time of onset and the magnitude of the trend are estimated for each of the two time series (numbers of failures and numbers of maintenance actions) independently. For each current time t the estimated mean cost for each policy to upgrade the subsystem at some future time is computed using the current estimates of the trend.

The column "Best upgrade time" in Appendix G presents the times to upgrade the subsystem which correspond to the maximum estimated mean cost advantage policies for each current time. If the maximum estimated mean cost advantage policy is never to upgrade (negative mean cost advantage), then the time to upgrade is set equal to the horizon time, H=180. These results are shown on the computer screen or contained in the output file. Also displayed is the estimated mean cost advantage of the best policy, standard deviation of the cost advantage, and the mean plus or minus two standard deviation bound for the best policy for each time t. If the best policy is never to upgrade, then the mean and standard deviation of the cost advantage are computed for the policy that starts to upgrade immediately.

We first describe the results obtained by using the Bayesian procedure. These results appear in subsection 2 of Section B. The best policy for current times 5-20 are either to upgrade immediately or to never upgrade. We can compare the data with the policies. The data vary somewhat before time 20, but there is no evidence of a trend occurring. Because of the variation of the data, the results are unstable. An apparent trend appears after time 20. The Bayesian procedure suggests this subsystem should be upgraded immediately for all the times after time 20 except time 46 and time 50. The declining data values at times 40 to 53 appear not to be able to overcome the increasing numbers before them. We can also check

the "cost advantage of upgrade" column. Even these costs can not accurately represent the real future cost advantage, but it still gives further information for the subsystem upgrade. At time 46 and time 50 the small negative mean cost advantage compared with others do not give strong evidence to maintain current subsystem. Figure 6 presents graphs obtained from the Bayesian assessment of the variability of the estimated mean cost advantage for the best policies from times 6 to 60 as described in Section B of Appendix E.

The results of the maximum likelihood procedure are presented in subsection 3. We can compare these results with those of the Bayesian procedure. The only difference in best policies occurs at time 50. It changes from never upgrade to upgrade immediately. We also compare the estimated mean cost advantage of the best policies with those computed by the Bayesian procedure. Again, the variability of the estimated mean cost advantage is larger than the variability in Bayesian procedure. Figure 7 presents graphs obtained from the bootstrap assessment of variability obtained by the maximum likelihood procedure of the cost advantage of upgrade $\hat{C}_O(t)-\hat{C}_N(\tau,t)$ for the best policies from t=6 to t=60; if the best policy at time t is never to upgrade, the cost advantage of the best policy is $\hat{C}_O(t)-\hat{C}_N(0,t)$. The numerical values of mean cost advantage and two standard deviation bounds appear after the best policies. Displayed is the mean cost advantage

$$m(B;\tau_0,60) = \frac{1}{100}\sum_{b=1}^{100}\left[\hat{C}_O(b;60) - \hat{C}_N(b;\tau_0,60)\right], \qquad (4.3)$$

and the mean plus and minus two standard deviations, where τ_0 is that time which maximizes the cost advantage for the original data; if the best policy is never to upgrade, then $\tau_0=0$. The bootstrap variance is

$$\xi^2(B;\tau_0,60) = \frac{1}{99}\sum_{b=1}^{100}\left[\hat{C}_O(b;60) - \hat{C}_N(b;\tau_0,60) - m(B;\tau_0,60)\right]^2. \qquad (4.4)$$

Displayed is

$$m(B;\tau_0,60) \pm 2\xi(B;\tau_0,60). \qquad (4.5)$$

Comparison of Figures 6 and 7 indicates that the bootstrap estimates of the variability of the estimated mean cost advantage of the best policies is larger than those for the Bayesian procedure. This larger variability for the maximum likelihood procedure is due to more variability in the bootstrap distribution of the estimated time of onset of subsystem degradation. The maximum likelihood procedure appears to be more sensitive to local features in the data than the Bayesian procedure.

The two standard deviation bounds appearing in Figure 6 and Figure 7 indicate that the variability of the estimated mean cost advantage becomes relatively small and stable after time 30. Notice that the estimated mean cost advantage is positive, which suggests that it is advantageous to start an

upgrading process. However, since the lower confidence bound is negative, it may still be worthwhile to wait for more evidence before starting the upgrading program.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Preliminary mathematical models have been formulated for the possible onset and growth of subsystem degradation. The model recognizes that the time of onset of a degrading trend may be random, and hence initially unknown, and that the trend magnitude is also initially unknown. The trend magnitude will become better known as more data is accumulated. Statistical procedures have been developed to estimate the time of onset and the trend magnitude. A cost model that is compatible with the existing decision aid, the ROI procedure, has been used to develop procedures (which recognize the uncertainty concerning the time of onset and magnitude) to determine estimated costs and the associated risks of upgrading the subsystem at different times in the future. An experiment using real data gives reasonable results and indicates that the consideration of variability in policy costs due to uncertainty concerning the time of onset and trend magnitude can lead to wiser decisions.

B. RECOMMENDATIONS

Two procedures to estimate the mean cost advantage and assess its variability are considered. One uses maximum likelihood for estimation and the bootstrap to assess

variability. The other uses a Bayesian model. The Bayesian procedure requires much less computational effort than the maximum likelihood/bootstrap procedure and appears to give similar results. Thus we suggest that the Bayesian procedure be used to estimate best time to upgrade and to assess variability of a cost advantage.

The changepoint model considered in this thesis has a linear trend after the changepoint. A linear trend may overestimate the magnitude of the degradation. Other possibilities exist. For example, another possible model is that the trend be proportional to the square root or some other power less than 1 of the time since the changepoint; that is,

$$\begin{aligned} X_i &\sim N(\mu, \sigma^2), & 0 \le i \le C; \\ &\sim N(\mu + \sqrt{i-C}\,\eta, \sigma^2), & C+1 \le i. \end{aligned} \tag{5.1}$$

Future work can extend the estimation procedures to such cases, and study the sensitivity of change policies and their costs to different specifications of degradation growth.

The cost of the subsystem in this thesis is the sum of costs due to failures and maintenance actions. In this thesis the costs due to failures and the costs due to maintenance actions are estimated separately. Future work can extend the estimation procedures to multivariate time series, if appropriate.

APPENDIX A

NORMAL RANDOM NUMBER GENERATION

A simple scheme for generating Normal random variables is the Box-Muller technique (G.S. Fishman 1978). The procedure generates two independent standardized Normally distributed variables X and Y as follows.

1. Generate U_1, U_2 as independent random variables uniformly distributed on (0,1).
2. Set

$$X = \sqrt{-2\,(InU_1)}\ \cos(2\pi U_2)$$
$$Y = \sqrt{-2\,(InU_1)}\ \sin(2\pi U_2)\,. \tag{A.1}$$

APPENDIX B

MAXIMUM LIKELIHOOD ESTIMATION

Suppose observations of the variables (numbers of failures or maintenance actions during time periods 1,...,t) X_1, X_2, ..., X_t are available; denote them by x_1, x_2, ..., x_t. Then the likelihood function for the unknown parameters, μ, C, η, σ^2 is as follows for the model in Chapter II (Donald P. Gaver and Patricia A. Jacobs 1992). Since the number of failures (maintenance actions) in successive time periods are assumed to be independent, for time $1 \le C \le t$, the likelihood function is

$$L(\mu, C, \eta, \sigma^2; data) = \prod_{i=1}^{C} \frac{e^{-(x_i-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \prod_{i=C+1}^{t} \frac{e^{-(x_i-\mu-(i-C)\eta)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \tag{B.1}$$

so the log-likelihood is

$$l(\mu, C, \eta, \sigma^2; data) = \sum_{i=1}^{C} \frac{-(x_i-\mu)^2}{2\sigma^2} + \sum_{i=C+1}^{t} \frac{-(x_i-\mu-(i-C)\eta)^2}{2\sigma^2} - \frac{t}{2}\ln\sigma^2 + constant. \tag{B.2}$$

This can be concisely written as

$$l(\mu, C, \eta, \sigma^2; data) = \sum_{i=1}^{t} \frac{-(x_i-\mu-(i-C)^+\eta)^2}{2\sigma^2} - \frac{t}{2}\ln\sigma^2 + constant. \tag{B.3}$$

where

$$(i-C)^{+} = \begin{cases} i-C & \text{if } i>C, \text{ and} \\ 0 & \text{if } i\le C. \end{cases} \qquad \textbf{(B.4)}$$

Note that the above applies if there is a changepoint within the range of observation; otherwise, if $C > t$ then

$$L(\mu, C, \eta, \sigma^2; data) = \prod_{i=1}^{t} \frac{e^{-(x_i-\mu)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \qquad \textbf{(B.5)}$$

and

$$l(\mu, C, \eta, \sigma^2; data) = \sum_{i=1}^{t} \frac{-(x_i-\mu)^2}{2\sigma^2} - \frac{t}{2}\ln\sigma^2 + constant. \qquad \textbf{(B.6)}$$

Now in the following hold C fixed and behave as if it were known and the objective is to maximize l with respect to μ, η, and σ^2. Begin by differentiating with respect to μ:

$$\begin{aligned} \frac{\partial l}{\partial \mu} &= \sum_{i=1}^{C} \frac{x_i-\mu}{\sigma^2} + \sum_{i=C+1}^{t} \frac{x_i-\mu-(i-C)\eta}{\sigma^2}, \quad \text{if } 0\le C\le t; \\ &= \sum_{i=1}^{t} \frac{x_i-\mu}{\sigma^2}, \quad \text{if } C>t. \end{aligned} \qquad \textbf{(B.7)}$$

These expressions can be simplified and combined:

$$\begin{aligned} \sigma^2 \frac{\partial l}{\partial \mu} &= t\bar{x}(t) - t\mu - \eta\sum_{j=1}^{t-C} j \\ &= t\bar{x}(t) - t\mu - \left(\frac{(t-C)^2+(t-C)}{2}\right)\eta, \quad \text{for } t\ge C \\ &= t\bar{x}(t) - t\mu, \quad \text{for } t<C \end{aligned} \qquad \textbf{(B.8)}$$

where

$$\overline{x}(t) = \frac{1}{t}\sum_{i=1}^{t} x_i. \tag{B.9}$$

Rewrite this as

$$\frac{\partial l}{\partial \mu} = t(\overline{x}(t)-\mu) - \left(\frac{((t-C)^+)^2 + (t-C)^+}{2}\right)\eta \tag{B.10}$$

where

$$(t-C)^+ = \begin{cases} t-C & \text{if } t \geq C; \\ 0 & \text{if } t < C. \end{cases} \tag{B.11}$$

If the derivative is set equal to zero we obtain the first "normal equation"

$$\mu + \psi_1(C,t)\eta = \overline{x}(t) \tag{B.12}$$

where here

$$\psi_1(C,t) = \frac{1}{t}\left(\frac{((t-C)^+)^2 + (t-C)^+}{2}\right). \tag{B.13}$$

Next differentiate (B.3) with respect to η:

$$\begin{aligned} \sigma^2 \frac{\partial l}{\partial \eta} &= \sum_{i=1}^{t} (x_i - \mu - (i-C)^+\eta)(i-C)^+ \\ &= \sum_{i=1}^{t} x_i (i-C)^+ - \mu \sum_{i=1}^{t} (i-C)^+ - \eta \sum_{i=1}^{t} ((i-C)^+)^2 \\ &= t\overline{x}_2(C,t) - \mu t \psi_1(C,t) + \eta t \psi_2(C,t) \end{aligned} \tag{B.14}$$

where

$$\bar{x}_2(C,t) = \frac{1}{t}\sum_{i=1}^{t} x_i (i-C)^+ \tag{B.15}$$

$$\psi_2(C,t) = \frac{1}{t}\sum_{i=1}^{t} ((i-C)^+)^2 = \frac{1}{t}\left(\frac{((t-C)^+)^3}{3} + \frac{((t-C)^+)^2}{2} + \frac{(t-C)^+}{6}\right). \tag{B.16}$$

Set the derivative equal to zero to obtain the second normal equation

$$\psi_1(C,t)\mu + \psi_2(C,t)\eta = \bar{x}_2(C,t)\,. \tag{B.17}$$

Differentiate with respect to σ^2

$$\frac{\partial l}{\partial \sigma^2} = \sum_{i=1}^{t} \frac{1}{2}(x_i - \mu - (i-C)^+\eta)^2 \left(\frac{1}{\sigma^2}\right)^2 - \frac{t}{2}\left(\frac{1}{\sigma^2}\right); \tag{B.18}$$

if this is set equal to zero and solved for σ^2 there results

$$\hat{\sigma}^2 = \frac{1}{t}\sum_{i=1}^{t} (x_i - \hat{\mu} - (i-C)^+\hat{\eta})^2 . \tag{B.19}$$

Now solve the first two normal equations for the maximum likelihood estimate, conditional on C; the result is:

$$\hat{\mu}(C) = \frac{\psi_2\bar{x} - \psi_1\bar{x}_2}{\psi_2 - (\psi_1)^2} \tag{B.20}$$

$$\hat{\eta}(C) = \frac{\bar{x}_2 - \psi_1\bar{x}}{\psi_2 - (\psi_1)^2} \tag{B.21}$$

for $C<t$; for $C \geq t$, $\hat{\mu}(C)=\bar{x}$, $\hat{\eta}(C)=0$. These can now be substituted into (B.19) to obtain the maximum likelihood estimate for σ^2 in terms of the other estimates, all conditional on the value

of C. Finally substitute the above estimates into the expression for the negative of the log likelihood:

$$
\begin{aligned}
S(C;data) &\equiv -\frac{2}{t}l(\hat{\mu}(C,t),C,\hat{\eta}(C,t),\hat{\sigma}^2(C,t);data) \\
&= \frac{1}{t}\sum_{i=1}^{C}\frac{(x_i-\hat{\mu}(C,t))^2}{\hat{\sigma}^2(C,t)} + \frac{1}{t}\sum_{i=C+1}^{t}\frac{(x_i-\hat{\mu}(C,t)-(i-C)\hat{\eta}(C,t))^2}{\hat{\sigma}^2(C,t)} \\
&\quad + \ln\hat{\sigma}^2(C,t) \\
&= 1 + \ln\hat{\sigma}^2(C,t) \qquad \textbf{(B.22)}
\end{aligned}
$$

and obtain the value of C that minimizes $S(C;data)$ over the range (1, 2, ..., t); denote this by $\hat{C}(t)$; the last equality in the above expression follows from the definition of $\hat{\sigma}^2(C,t)$ given by (B.19). Thus, the estimate of C is chosen to minimize the sum of the squared residuals. If the minimum of $S(C;data)$ occurs at $t=C$, then the conclusion is that no change has occurred in $[1,t]$. Note that all estimated parameter values, namely $\hat{\mu}$, $\hat{\eta}$, and $\hat{\sigma}^2$ depend upon the C value in use, and so the dependence of S upon C involves that implicit dependency. Once $\hat{C}(t)$ is developed this value is substituted into the expressions for $\hat{\mu}$, $\hat{\eta}$, and $\hat{\sigma}^2$ to obtain the maximum likelihood estimates of those parameters.

APPENDIX C

BAYESIAN ESTIMATION

An enhanced version of the basic model presented in Appendix B is obtained by assuming that the changepoint (time of onset of degradation) is a random variable, C, with specified distribution whose parameter is unknown and subject to a probability density, $\pi(\cdot)$. Specifically, suppose

$$P\{C=k\} = (1-p)^{k-1}p, \tag{C.1}$$

i.e. is geometric, and that the parameter p has a prior distribution $\pi(\cdot)$.

We also use the linear normal model here. Putting (uninformative) priors on μ, η, and p, it is shown that the joint posterior density of those is straightforwardly obtained; the parameter σ^2 is initially estimated from residuals. In principle all of the above could be carried for any arbitrary, but reasonable, discrete distribution that might better represent what is known about the changepoint process. A similar statistical model was used by Smith (1975).

In what follows we sketch the development (Donald P. Gaver and Patricia A. Jacobs 1992). Suppose that observations, $x_1, \ldots, x_t$, are available up to time t, it follows that

$$P\{p\in(dp),\, C=k,\, \mu\in(d\mu),\, \eta\in(d\eta)\,|X_1=x_1,\,\ldots,\,X_i=x_i,\,\ldots,\,X_t=x_t,\,\sigma^2\}$$

$$= \pi(p)(1-p)^{k-1}p\prod_{i=1}^{t}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(x_i-\mu-(i-k)^+\eta)^2\right\}d\mu\, d\eta\, dp \quad \textit{for } k\le t$$

$$= \pi(p)(1-p)^{k-1}\prod_{i=1}^{t}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(-x_i-\mu)^2\right\}d\mu\, d\eta\, dp \quad \textit{for } k=t+1 \quad \textbf{(C.2)}$$

where

$$(i-k)^+ = \begin{cases} i-k & \textit{if } i\ge k, \textit{ and} \\ 0 & \textit{if } i<k. \end{cases} \qquad \textbf{(C.3)}$$

The term involving $(1-p)^t$ represents the case in which no changepoint has occurred; we will set $k=t+1$ for this case.

By a completion-of-squares process one can write the likelihood function for given $C=k$ as a bivariate normal density with parameters dependent on k and data up to t; the exponential term of the likelihood is written as

$$\prod_{i=1}^{t}\exp\left\{-\frac{1}{2\sigma^2}(x_i-\mu-(i-k)^+\eta)^2\right\} \qquad \textbf{(C.4)}$$

$$= c\,\exp\left\{-\left(\frac{1}{2(1-\rho^2)}\left(\frac{(\mu-\bar{\mu})^2}{\gamma^2}-2\rho\frac{(\mu-\bar{\mu})(\eta-\bar{\eta})}{\gamma\nu}+\frac{(\eta-\bar{\eta})^2}{\nu^2}\right)-K(k,t)\right\}$$

for $1\le k\le t-1$; for $k\ge t$ we have no changepoint so the exponential term of the likelihood is of the form

$$\prod_{i=1}^{t} \exp\left\{-\frac{1}{2\sigma^2}(x_i-\mu)^2\right\}$$

$$= c\,\exp\left\{-\frac{1}{2}\frac{(\mu-\bar{\mu})^2}{\gamma^2} - K(t,t)\right\} \tag{C.5}$$

where in the above c is a constant, and the parameters all depend upon k, t, and $\underline{x}(t)$, the data up to time t.

For k<t, the parameters of the bivariate normal (C.4) turn out to be

$$\bar{\mu}(k,t) = \frac{\bar{x}(t)\psi_2 - \bar{x}_2(k,t)\psi_1}{\psi_2 - (\psi_1)^2}; \tag{C.6}$$

$$\bar{\eta}(k,t) = \frac{\bar{x}_2(k,t) - \bar{x}(t)\psi_1}{\psi_2 - (\psi_1)^2}; \tag{C.7}$$

$$\gamma^2(k,t) = \frac{\psi_2}{\psi_2 - (\psi_1)^2}\frac{\sigma^2}{t}; \tag{C.8}$$

$$\nu^2(k,t) = \frac{1}{\psi_2 - (\psi_1)^2}\frac{\sigma^2}{t}; \tag{C.9}$$

and

$$\rho(k,t) = -\frac{\psi_1}{\sqrt{\psi_2}}; \tag{C.10}$$

where

$$\overline{x}(t) = \frac{1}{t}\sum_{i=1}^{t} x_i; \tag{C.11}$$

$$\overline{x}_2(k,t) = \frac{1}{t}\sum_{i=1}^{t} x_i (i-k)^+; \tag{C.12}$$

$$\psi_1 = \frac{1}{t}\sum_{i=1}^{t} (i-k)^+; \tag{C.13}$$

$$\psi_2 = \frac{1}{t}\sum_{i=1}^{t} ((i-k)^+)^2; \tag{C.14}$$

and

$$K(k,t) = \frac{1}{2\sigma^2}\sum_{i=1}^{t}\left(x_i - \overline{\mu}(k,t) - \overline{\eta}(k,t)(i-k)^+\right)^2; \tag{C.15}$$

For the case $k \geq t$

$$\overline{\mu}(k,t) = \frac{1}{t}\sum_{i=1}^{t} x_i; \tag{C.16}$$

$$\gamma^2(k,t) = \frac{\sigma^2}{t}; \tag{C.17}$$

and

$$K(k,t) = \frac{1}{2\sigma^2}\sum_{i=1}^{t}\left(x_i - \overline{\mu}(k,t)\right)^2; \tag{C.18}$$

$\overline{\eta}(k,t)=0$, $\upsilon^2(k,t)=0$, and $\rho(k,t)=0$. These values can be derived directly from (C.4) and (C.5); the procedure is similar to that in Appendix B.

If the bivariate normal form is utilized in (C.2) and the integration is performed over p we obtain the joint conditional density of C, μ, and η given the data and σ^2 in the form

$$P\{C=k, \mu\in(d\mu), \eta\in(d\eta)\,|\,\underline{x}(t), \sigma^2\} =$$

$$\frac{\pi^*(k,t)}{2\pi\sqrt{1-\rho^2}\gamma\nu}\exp\left[-\frac{1}{2(1-\rho^2)}\left(\frac{(\mu-\overline{\mu})^2}{\gamma^2}-2\rho\frac{(\mu-\overline{\mu})(\eta-\overline{\eta})}{\gamma\nu}+\frac{(\eta-\overline{\eta})^2}{\nu^2}\right)\right] \quad \textbf{(C.19)}$$

for $k<t$ where

$$\pi^*(k,t) = c^*\int_0^1(1-p)^{k-1}p\pi(p)\,dp\exp(-K(k,t))(2\pi\sqrt{1-\rho^2}\gamma\nu); \quad \textbf{(C.20)}$$

for $k\geq t$

$$P\{C=k, \mu\in(d\mu)\,|\,\underline{x}(t), \sigma^2\} = \pi^*(k,t)\frac{1}{\sqrt{2\pi}\gamma}\exp\left(-\frac{1}{2\gamma^2}(\mu-\overline{\mu})^2\right) \quad \textbf{(C.21)}$$

with

$$\pi^*(t,t) = c^*\int_0^1(1-p)^{t-1}p\pi(p)\,dp\exp(-K(t,t))(\sqrt{2\pi}\gamma) \quad \textbf{(C.22)}$$

$$\pi^*(t+1,t) = c^*\int_0^1(1-p)^{t}\pi(p)\,dp\exp(-K(t+1,t))(\sqrt{2\pi}\gamma) \quad \textbf{(C.23)}$$

and

$$c^* = \left[\begin{array}{l} \sum_{k=1}^{t-1}\int_0^1 (1-p)^{k-1} p\pi(p)\,dp \exp(-K(k,t))(2\pi\sqrt{1-\rho^2}\gamma\nu) \\ \quad + \int_0^1 (1-p)^{t-1} p\pi(p)\,dp \exp(-K(t,t))(\sqrt{2\pi}\gamma) \\ \quad + \int_0^1 (1-p)^{t}\pi(p)\,dp\ \exp(-K(t+1,t))(\sqrt{2\pi}\gamma) \end{array}\right]^{-1} \qquad \textbf{(C.24)}$$

Note that $\{\pi^*(k,t),\ k\le t\}$ is the marginal probability that the changepoint occurs at any time k up to and including t; while $\pi^*(t+1,t)$ is the posterior probability that no changepoint has occurred up to time t.

For each time t, the estimate of σ^2 is computed from the squared residuals for each possible value of $C=k$ in the following manner; let

$$\hat{\sigma}^2(k,t) = \frac{1}{t-1}\sum_{i=1}^{t}\left(x_i-\bar{\mu}(k,t)-\bar{\eta}(k,t)(i-k)^+\right)^2 \quad \text{if } k\le t; \qquad \textbf{(C.25)}$$

$$\hat{\sigma}^2(k,t) = \frac{1}{t-1}\sum_{i=1}^{t}\left(x_i-\bar{\mu}(k,t)\right)^2 \quad \text{if } k>t. \qquad \textbf{(C.26)}$$

Finally, the estimate of the variance σ^2 based on data $x_1,\ldots,$ x_t is

$$\hat{\sigma}^2(t) = \sum_{k=1}^{t+1}\pi^*(k,t)\,\hat{\sigma}^2(k,t)\,. \qquad \textbf{(C.27)}$$

Given $C=k$, $k<t$, and the data $x_1,\ldots,$ x_t, the posterior distribution of (μ,η) is bivariate normal with mean $(\bar{\mu}(k,t),\bar{\eta}(k,t))$, variance of μ equal to $\gamma^2(k,t)$, variance of

η equal to $\upsilon^2(k,t)$, and correlation $\rho(k,t)$; for $k=t,\ t+1$, $\eta=0$ and the posterior distribution of μ is normal with mean $\bar{\mu}(k,t)$ and variance $\gamma^2(k,t)$. Hence, given the data $x_1,\ldots,\ x_t$, the posterior distribution of (μ,η) is a mixture of bivariate normal distributions with mixture distribution $\{\pi^*(k,t),\ k\le t+1\}$.

Since the bivariate normal has 5 parameters to be estimated, the estimation procedure begins with data $x_1,\ldots,\ x_6$. The initial estimate of σ^2 is

$$\hat{\sigma}^2 = \frac{1}{4}\sum_{i=1}^{5}(x_i-\bar{x})^2 \qquad \textbf{(C.28)}$$

where $\bar{x}$ is the sample average of the first 5 data points. For each time t, estimates of the posterior distribution are obtained from equations (C.6)-(C.26). The updated estimate of σ^2 is used as input for the calculations for the next time period.

APPENDIX D

A MATHEMATICAL MODEL FOR COST AND RETURN-ON-INVESTMENT

In this Appendix we describe the cost model. The cost model is formulated to reflect the costs that are used in the decision aid in current use, the ROI procedure.

There are several different unit costs. There is a cost due to subsystem failures; a cost due to subsystem maintenance action; and a cost due to AV-DLR action.

In addition there is a planned horizon H during which the parent system will be operative; when the horizon is reached all (remaining) parents are stored or disposed of. There is also a lead time L before the upgrade is initiated and an installation period of length J.

Section A below describes the calculation of the unit costs. Section B below describes the cost model in detail. Section C describes the cost estimation procedure.

Let $\hat{C}_N(\tau, t)$ denote the estimated mean cost of deciding at time t to begin the procedure to upgrade the subsystem τ time units in the future. Let $\hat{C}_O(t)$ denote the cost of deciding never to upgrade. If

$$\min_{\tau} \; \hat{C}_N(\tau, t) < \hat{C}_O(t) \tag{D.1}$$

then it may be advantageous to begin upgrading process at that time τ which minimizes the left hand side of (D.1). However,

a large amount of uncertainty concerning $\hat{C}_N(\tau,t)$ and $\hat{C}_O(t)$ may indicate that it is better to postpone the decision to upgrade until more data has been obtained. Appendix E describes procedures to assess the uncertainty of the cost estimates.

A. UNIT COSTS

This section describes the computation of the unit costs. All the parameters used in the computation of cost can be obtained from the ROI program. These costs for a radar transmitter for the F-14A can be found in the Appendix F. There are 5 costs in the ROI procedure for both the current and upgraded subsystem. The 5 costs are computed as follows

1. Using the "O" MH/MA times the MA at "O" level (MA/ML1 Ratio) gives the total number of hours ("O" MH) spent by the squadrons on maintaining this subsystem. Multiplying the "O" MH by the composite rate for "O" level (ML1 Rate) gives the "O" level manpower cost.

2. Similarly, using the "I" MH/MA times the MA at "I" level (MA/ML2 Ratio) gives the total number of hours ("I" MH) spent by the squadrons on maintaining this subsystem. Multiplying the "I" MH by the composite rate for "I" level (ML2 Rate) gives the "I" level manpower cost.

3. Again using the simple relationship of MH/F times the VF gives the total number of hours spend on repairing this subsystem. Multiplying the hours by the composite rate for "I" level (ML2 Rate) gives the manpower cost per repair.

4. The cost of materials used for repairs is the product of cost of materials per repair ($/Rpr) times the number of repairs.

5. The total AVDLR cost is the product of the number of BCM systems times the AVDLR unit cost.

The calculation of the model unit cost parameters due to failure, maintenance action, and AV-DLR action is summarized as follows

$$\begin{aligned} C_{OM} = {} & (\text{"O" } MH/MA)_{old} (MA/ML1\ Ratio)(ML1\ Rate) \\ & + (\text{"I" } MH/MA)_{old} (MA/ML2\ Ratio)(ML2\ Rate), \end{aligned} \tag{D.2}$$

$$\begin{aligned} C_{NM} = {} & (\text{"O" } MH/MA)_{new} (MA/ML1\ Ratio)(ML1\ Rate) \\ & + (\text{"I" } MH/MA)_{new} (MA/ML2\ Ratio)(ML2\ Rate), \end{aligned} \tag{D.3}$$

$$C_{OF} = (MH/F)_{old} (ML2\ Rate) + (\$/Rpr)_{old}, \tag{D.4}$$

$$C_{NF} = (MH/F)_{new} (ML2\ Rate) + (\$/Rpr)_{new}, \tag{D.5}$$

$$C_{OA} = (AVDLR\ Cost/Unit)_{old}, \tag{D.6}$$

$$C_{NA} = (AVDLR\ Cost/Unit)_{new}. \tag{D.7}$$

The subscript O represents current subsystem; N represents upgraded subsystem; F represents failure; M represents maintenance action; and A represents AV-DLR action.

B. ESTIMATED FUTURE MEAN COST

Fix a time t and let $\hat{C}_F$ (respectively $\hat{C}_M$) be the estimate obtained at that time of the time of onset of subsystem

degradation due to failures (respectively maintenance actions); let β_F (respectively β_M) be the estimate of the constant mean number of failures (respectively maintenance actions) in each time period before the onset of subsystem degradation; and let η_F (respectively η_M) be the estimate of the magnitude of the linear degrading trend in the mean number of failures (respectively maintenance actions) after the onset of subsystem degradation.

The estimated future mean (total, undiscounted) cost due to subsystem failures of a policy that initiates upgrading τ time units in the future is

$$\hat{C}_{NF}(\tau,t) = \begin{cases} c_{OF}\beta_F\left[(\tau+L+1)+\sum_{s=1}^{J}(1-\alpha(s))\right] \\ \quad + c_{NF}\lambda_F\left[\sum_{s=1}^{J}\alpha(s)+(H-(t+\tau+L+J))\right] & \text{if } \hat{C}_F>t \\ c_{OF}\left[\begin{array}{l}\sum_{s=0}^{L+\tau}(\beta_F+\eta_F(s+(t-\hat{C}_F)^+)) \\ +\sum_{s=L+\tau+1}^{L+\tau+J}(\beta_F+\eta_F(s+(t-\hat{C}_F)^+))(1-\alpha(s-(\tau+L)))\end{array}\right] \\ \quad + c_{NF}\lambda_F\left[\sum_{s=1}^{J}\alpha(s)+(H-(t+\tau+L+J))\right] & \text{if } \hat{C}_F\le t, \end{cases} \qquad \textbf{(D.8)}$$

where L is the lead time to begin installation of the upgraded subsystem; λ_F is the mean number of failures per month for the upgraded subsystem; J is the length of the installation period; $\alpha(s)$ is the fraction of subsystems that have been

upgraded s time units into the installation period; and H is the time horizon, (the useful lifetime of the subsystem).

The estimated future mean cost due to subsystem maintenance actions of a policy that switches to the upgraded subsystem τ time units in the future is

$$\hat{C}_{NM}(\tau,t) = \begin{cases} c_{OM}\beta_M\left[(\tau+L+1)+\sum_{s=1}^{J}(1-\alpha(s))\right] \\ \quad + c_{NM}\lambda_M\left[\sum_{s=1}^{J}\alpha(s)+(H-(t+\tau+L+J))\right] & \text{if } \hat{C}_M > t \\ c_{OM}\left[\begin{array}{l}\sum_{s=0}^{L+\tau}(\beta_M+\hat{\eta}_M(s+(t-\hat{C}_M)^+)) \\ +\sum_{s=L+\tau+1}^{L+\tau+J}(\beta_M+\hat{\eta}_M(s+(t-\hat{C}_M)^+))(1-\alpha(s-(\tau+L)))\end{array}\right] \\ \quad + c_{NM}\lambda_M\left[\sum_{s=1}^{J}\alpha(s)+(H-(t+\tau+L+J))\right] & \text{if } \hat{C}_M \le t. \end{cases} \tag{D.9}$$

The total estimated future mean cost of the policy that begins the upgrading process τ time units in the future is

$$\begin{aligned} \hat{C}_N(\tau,t) &= \hat{C}_{NF}(\tau,t) + \hat{C}_{NM}(\tau,t) + c_F \\ &\quad + c_{OA}\gamma_A\left((\tau+L+1)+\sum_{s=1}^{J}(1-\alpha(s))\right) \\ &\quad + c_{NA}\lambda_A\left(\sum_{s=1}^{J}\alpha(s)+(H-(t+\tau+L+J))\right) \end{aligned} \tag{D.10}$$

where c_F is the initial fixed cost for the upgrade; γ_A is the mean number of AV-DLR action per month for current subsystem and is computed as the average number of BCM's for the last 24

month; and λ_A is the assumed mean number of AV-DLR actions per month for upgraded subsystem.

The estimated mean cost due to failures of a policy that never upgrades the subsystem is

$$\hat{C}_{OF}(t) = \begin{cases} c_{OF}\hat{\mu}_F(H-(t-1)) & \text{if } \hat{C}_F > t \\ c_{OF}\left[\sum_{s=0}^{H-t} (\hat{\mu}_F + \hat{\eta}_F(s+(t-\hat{C}_F)^+))\right] & \text{if } \hat{C}_F \leq t. \end{cases} \qquad \textbf{(D.11)}$$

The estimated cost due to maintenance actions of a policy that never upgrades the subsystem is

$$\hat{C}_{OM}(t) = \begin{cases} c_{OM}\hat{\mu}_M(H-(t-1)) & \text{if } \hat{C}_M > t \\ c_{OM}\left[\sum_{s=0}^{H-t} (\hat{\mu}_M + \hat{\eta}_M(s+(t-\hat{C}_M)^+))\right] & \text{if } \hat{C}_M \leq t. \end{cases} \qquad \textbf{(D.12)}$$

The total estimated mean cost incurred by a policy that never upgrades the subsystem is

$$\hat{C}_O(t) = \hat{C}_{OF}(t) + \hat{C}_{OM}(t) + c_{OA}\gamma_A(H-(t-1)). \qquad \textbf{(D.13)}$$

C. THE ESTIMATION PROCEDURE FOR COSTS

Table 1 in the Appendix F lists the menu of the ROI procedure. All of the cost parameters for the current subsystem and all parameters for the upgraded subsystem used in our decision aid appear in, or are computed from, the numbers in Table 1.

The current decision aid, the ROI program, uses the MTBF and the MTBMA listed in Table 1; these are obtained by averaging the measures of performance for the last 24 months in the MFHBF and MFHBMA time-series data which appear in Table 3. The decision aid developed in this thesis uses all MFHBF and MFHBMA time-series data. Our decision aid uses the estimated measures of performance for the upgrade subsystem appearing in the improved column of Table 1. The cost for the upgrade and the length of the lead time are also taken from Table 1. The other information section of Table 1 contains the cross-over month computed by the ROI decision aid. The cross over month is the ROI program's measure of evaluating the cost effectiveness of the upgrade; it is month in which the total cost of subsystem upgrade becomes smaller than the estimated cost of not upgrading the current subsystem. The other information section also contains the number of systems and their use per month; these are computed as average values of measures for the last 24 months in Table 2 which gives the total number of aircraft and the total flight hours per month. The other four values in the section are also computed by ROI procedure.

Because the total number of aircraft and total flight hours per month change over time, we use the mean number of failures instead of the actual number of failures in each month to estimate the performance of the subsystem. This value is the mean monthly flight hours, computed as the average

number of systems times the average use per month divided by the mean flight hours between failures (column MFHBF in Table 3) for each month. The mean number of maintenance actions each month is estimated as the same mean monthly flight hours as above divided by the mean flight hours between maintenance actions (column MFHBMA in Table 3).

For each time t the procedure of Appendices B and C are used to independently estimate $(\beta_F, \eta_F, \hat{C}_F)$ and $(\beta_M, \eta_M, \hat{C}_M)$. The parameter γ_A is obtained by taking average of BCM's for the last 24 months. The number of time periods used for the installation, J, is obtained as the number of systems in Table 1 divided by the installation rate (Kits/Mth installed) in the same table and rounded up to the next integer. This integer is then compared to the time remaining (the starting time for the upgrade until the time horizon, $H-t-L-\tau$), the smaller of these two numbers is then used to represent J. The fraction of old subsystems that have been upgraded s time periods into the installation period, $\alpha(s)$, is obtained as s times the installation rate divided by the number of systems. The fixed cost to upgrade c_F is obtained by adding the cost per kit and the cost to install one kit together multiplied by the number of systems, and added to the other terms in the section "Cost for fix". The lead time L appears in the lead time section. It is obtained by adding all the items in the section except "Kits/Mth installed".

APPENDIX E

ASSESSMENT OF UNCERTAINTY

In this Appendix we discuss procedures to assess the variability of the cost estimates described in chapter II.

A. THE BOOTSTRAP PROCEDURE

A re-sampling technique called *the bootstrap* can be used to assess the variability of the maximum likelihood estimated mean cost associated with a policy (B. Efron and R. Tibshirani 1986).

Fix a time t and let $\hat{\mu}_j(t)$, $\hat{\eta}_j(t)$, $\hat{\sigma}_j(t)$, and $\hat{C}_j(t)$, $j \in \{F,M\}$ denote the maximum likelihood estimates obtained from the data. A bootstrap replication has the following steps.

1. Using model (2.1) with parameter values equal to the estimates $\hat{\mu}_F(t)$, $\hat{\eta}_F(t)$, $\hat{\sigma}_F(t)$, and $\hat{C}_F(t)$ simulate data $x_{F1}(b)$, $x_{F2}(b), \ldots, x_{Ft}(b)$. Using the simulated data use the maximum likelihood procedure to obtain bootstrap estimates $\hat{\mu}_F(b,t)$, $\hat{\eta}_F(b,t)$, $\hat{\sigma}_F(b,t)$, and $\hat{C}_F(b,t)$; b denotes the $b^{\underline{th}}$ bootstrap simulation; b=1, 2, ..., B, where B is the number of bootstrap samples utilized.

2. Repeat step 1 for the estimates $\hat{\mu}_M(t)$, $\hat{\eta}_M(t)$, $\hat{\sigma}_M(t)$, and $\hat{C}_M(t)$ to obtain bootstrap estimates $\hat{\mu}_M(b,t)$, $\hat{\eta}_M(b,t)$, $\hat{\sigma}_M(b,t)$, and $\hat{C}_M(b,t)$.

3. Using the bootstrap estimates obtained in steps 1 and 2 above compute the future mean cost of a policy that switches to a new subsystem τ time units into the future, $\hat{C}_N(b;t,\tau)$ using (D.10). Also compute the cost of never changing $\hat{C}_O(b;t)$ from (D.13).

4. Compute the cost advantage of upgrade

$$\hat{C}_O(b;t) - \hat{C}_N(b;\tau,t). \tag{E.1}$$

The results reported in Appendix G use B=100 bootstrap replications. After the 100 replication are generated, the mean and variance of the bootstrap cost advantage are computed; that is

$$m(B;\tau,t) = \frac{1}{100}\sum_{b=1}^{100}\left[\hat{C}_O(b;t) - \hat{C}_N(b;\tau,t)\right], \tag{E.2}$$

$$\xi^2(B;\tau,t) = \frac{1}{99}\sum_{b=1}^{100}\left[\hat{C}_O(b;t) - \hat{C}_N(b;\tau,t) - m(B;\tau,t)\right]^2. \tag{E.3}$$

B. THE BAYESIAN ASSESSMENT OF UNCERTAINTY

For series data x_{j1}, $x_{j2}, \ldots,$ x_{jt} the Bayesian procedure described in Appendix C yields a posterior distribution for the time of onset of subsystem degradation as of time t, namely $\hat{C}_j(t)$; it also gives estimates of the conditional variance of $\overline{\mu}_j(k,t)$, $\overline{\eta}_j(k,t)$, and their conditional covariance given $\hat{C}_j(t)=k$, for $j\in\{F,M\}$. These estimates together with the

cost model can be used to obtain the variance of the cost advantage for a policy.

For example for k<t

$$
\begin{aligned}
&E[\hat{C}_{OF}(t)-\hat{C}_{NF}(\tau,t)\,|C_F=k] \\
&= c_{OF}\left[\begin{array}{l}\overline{\mu}_F(k,t)\left((H-t-\tau-L)-\sum_{s=L+\tau+1}^{L+\tau+J}(1-\alpha(s-(L+\tau)))\right) \\ +\ \overline{\eta}_F(k,t)\left(\sum_{s=L+\tau+1}^{H-t}(s+(t-k)^{+})-\sum_{s=L+\tau+1}^{L+\tau+J}(s+(t-k)^{+})(1-\alpha(s-(L+\tau)))\right)\end{array}\right] \\
&\quad - c_{NF}\lambda_F\left[\sum_{s=1}^{J}\alpha(s)+(H-(t+\tau+L+J))\right] \\
&\equiv A(k;\tau,t)\overline{\mu}_F(k,t) + B(k;\tau,t)\overline{\eta}_F(k,t) - C(k;\tau,t) \qquad \textbf{(E.4)}
\end{aligned}
$$

and

$$
\begin{aligned}
&Var[\hat{C}_{OF}(t)-\hat{C}_{NF}(\tau,t)\,|C_F=k] \\
&= A(k;\tau,t)^2\,\gamma_F^2(k,t) + B(k;\tau,t)^2\,\nu_F^2(k,t) \qquad \textbf{(E.5)} \\
&\quad + 2\rho(k,t)\,\gamma_F(k,t)\,\nu_F(k,t)\,A(k;\tau,t)\,B(k;\tau,t)\,.
\end{aligned}
$$

Let

$$
m_F(\tau,t) = \sum_{k=1}^{t+1}\pi^{*}(k,t)\,E[\hat{C}_{OF}(t)-\hat{C}_{NF}(\tau,t)\,|C_F=k]\,, \qquad \textbf{(E.6)}
$$

then

$$Var[\hat{C}_{OF}(t) - \hat{C}_{NF}(\tau, t)]$$

$$= \sum_{k=1}^{t+1} \pi^*(k, t)\, Var[\hat{C}_{OF}(t) - \hat{C}_{NF}(\tau, t) \,|\, C_F = k] \tag{E.7}$$
$$+ \sum_{k=1}^{t+1} \pi^*(k, t)\big[E[\hat{C}_{OF}(t) - \hat{C}_{NF}(\tau, t) \,|\, C_F = k] - m_F(\tau, t)\big]^2 .$$

Finally, the mean of the total cost advantage is

$$m(\tau, t) = m_F(\tau, t) + m_M(\tau, t) - c_F$$
$$+ c_{OA}\gamma_A \left((H - t - \tau - L) - \sum_{s=1}^{J} (1 - \alpha(s)) \right) \tag{E.8}$$
$$- c_{NA}\lambda_A \left(\sum_{s=1}^{J} \alpha(s) + (H - (t + \tau + L + J)) \right),$$

and variance of the total cost advantage is

$$Var[\hat{C}_O(t) - \hat{C}_N(\tau, t)]$$
$$= Var[\hat{C}_{OF}(t) - \hat{C}_{NF}(\tau, t)] \tag{E.9}$$
$$+ Var[\hat{C}_{OM}(t) - \hat{C}_{NM}(\tau, t)]$$

since the other terms in the cost function are constant terms and we are assuming the failure time series and maintenance action time series are independent.

APPENDIX F

DATA SAMPLE

This Appendix contains a sample of the input data for both the decision aid developed in this thesis, the PASCAL program FCT.PAS and the decision aid currently in use, the ROI program. The data are for a radar transmitter for the F-14A. They are listed in the following tables. The ROI data is used directly in the program. The rest of the data sets are used to support the ROI data.

Table 1 presents data used in the ROI program. All the values for the current subsystem except "$/Rpr" and "AVDLR Cost/Unit" are computed by taking average values of these measures for the last 24 months. They are computed by the ROI program automatically. The values for the improved subsystem, the cost for fix, and the lead time sections of the table are input by the analyst for the upgraded subsystem. The "other information" section of the table is computed by the ROI program. Table 2 presents flight hours and numbers of aircraft for each month in a 5 year period. Table 3 presents the number of BCM's, the number of maintenance actions, the mean flight hours between failures, the mean flight hours between maintenance actions, and the number of failures for each month. Table 4 contains the man-hours for failure, for

organizational level maintenance action, and for intermediate level maintenance action for each month.

A. ROI DATA

TABLE 1. ROI DATA

TEC/TMS: AFWA/F-14A WUC: 74A1500 - T1224/AWG9 RADAR TRANSMITTER 0		
Statistics	Current	Improved
MTBF	42.47	80.00
MTBMA	20.60	20.60
"O" MH/MA	6.34	9.01
"I" MH/MA	9.01	9.01
MH/F	13.13	13.13
$/Rpr [Matl]	1082.64	1082.64
#BCMs[1-8]/Mth	7.80	4.00
AVDLR Cost/Unit	1120.36	1120.36
Cost for Fix		$
Non-recurring Engineering		1000000
Publications		250000
Cost per Kit		5000
Cost to Install one kit		2500
Cost of Spares		400000
Cost for Training		400000
Cost for Support Equipment		350000
Lead Time		Mths
Funds		24
Engineering		15
Kits		18
Start Instl of Kits		3
Total Lead Time		60

Kits/Mth Installed	15(Kits)
Other Information	
Cross over month	96
Number of systems	400
Use per month	25
ML1 Rate	15.28
ML2 Rate	18.35
MA/ML1 Ratio	0.92
MA/ML2 Ratio	0.47

MTBF (or MFHBF) - Mean time between failures

MTBMA (or MFHBMA) - Mean time between maintenance actions

"O" MH/MA - Organizational level maintenance Man-hours per maintenance action

"I" MH/MA - Intermediate level maintenance Man-hours per maintenance action

MH/F - Maintenance Man-hours per failure

$/Rpr [Matl] - Cost of material (bit & piece cost) per repair

BCMs[1-8]/Mth - Average monthly number of BCMs categories 1-8

AVDLR Cost/Unit - Unit cost per AV-DLR action

B. FLIGHT HOURS AND NUMBER OF AIRCRAFT DATA

TABLE 2. FLIGHT HOURS AND NUMBER OF AIRCRAFT

FLT-HRS/TOT-ACFT AFWA F-14A		
Data	FLT-HRS	TOT-ACFT
1991/12	4241.0	310
1991/11	5330.3	321
1991/10	6526.7	323
1991/09	7106.7	326
1991/08	7888.8	323
1991/07	5860.1	340
1991/06	6671.9	337
1991/05	9773.0	405
1991/04	8583.7	402
1991/03	10071.5	404
1991/02	15102.9	391
1991/01	13338.9	408
1990/12	8402.2	412
1990/11	8554.1	407
1990/10	10671.5	419
1990/09	10002.4	427
1990/08	10108.3	438
1990/07	8131.2	441
1990/06	9264.7	441
1990/05	9856.3	446
1990/04	10048.7	464
1990/03	10250.4	459
1990/02	9029.5	472
1990/01	10038.2	467
1989/12	6806.9	466

1989/11	7668.7	463
1989/10	11483.0	458
1989/09	9363.5	462
1989/08	10456.5	462
1989/07	9633.0	468
1989/06	10156.0	458
1989/05	10856.9	458
1989/04	10470.5	452
1989/03	10330.5	452
1989/02	9970.1	447
1989/01	9870.6	444
1988/12	8899.5	440
1988/11	9560.8	439
1988/10	9253.2	441
1988/09	10367.0	443
1988/08	10113.0	447
1988/07	8804.3	442
1988/06	10438.3	448
1988/05	10646.1	456
1988/04	10376.2	449
1988/03	10666.8	451
1988/02	10089.6	445
1988/01	9224.5	444
1987/12	6535.9	443
1987/11	8979.7	449
1987/10	9502.1	441
1987/09	9631.6	434
1987/08	9437.9	437
1987/07	9071.6	438
1987/06	10198.4	439

1987/05	9445.1	434
1987/04	9669.5	442
1987/03	9487.1	439
1987/02	9157.3	433
1987/01	8126.3	438

C. BCM, MA, MFHBF, MFHBMA, VF DATA

TABLE 3. BCM, MA, MFHBF, MFHBMA, VF

BCM/MA/MFHBF/MFHBMA/VF AFWA F-14A 74A1500 - T1224/AWG9 RADAR TRANSMITTER 0					
Data	BCM	MA	MFHBF	MFHBMA	VF
1991/12	3	204	46.6	20.8	91
1991/11	4	352	34.2	15.1	156
1991/10	9	362	36.2	18.0	180
1991/09	6	355	36.8	20.0	193
1991/08	15	429	35.7	18.4	221
1991/07	6	359	34.3	16.3	171
1991/06	11	378	35.3	17.7	189
1991/05	1	418	47.9	23.4	204
1991/04	6	382	42.5	22.5	202
1991/03	10	463	43.6	21.8	231
1991/02	12	567	50.3	26.6	299
1991/01	21	652	39.8	20.5	335
1990/12	12	392	42.9	21.4	196
1990/11	3	443	41.5	19.3	206
1990/10	14	441	51.1	24.2	209
1990/09	4	404	49.0	24.8	204
1990/08	9	500	39.8	20.4	256
1990/07	9	438	40.7	18.6	200
1990/06	5	401	47.8	23.1	194

1990/05	4	444	49.8	22.2	198
1990/04	1	433	51.8	23.2	194
1990/03	14	515	40.2	19.9	255
1990/02	3	489	43.8	18.5	206
1990/01	13	612	37.0	16.4	271
1989/12	10	379	43.6	18.0	156
1989/11	9	432	42.8	17.8	179
1989/10	15	597	45.2	19.2	254
1989/09	9	503	43.1	18.6	217
1989/08	13	559	38.4	18.7	272
1989/07	7	448	45.4	21.5	212
1989/06	10	476	40.5	21.3	251
1989/05	6	444	44.7	24.5	243
1989/04	12	542	37.8	19.3	277
1989/03	8	491	40.5	21.0	255
1989/02	7	417	47.5	23.9	210
1989/01	9	407	41.8	24.3	236
1988/12	8	370	45.4	24.1	196
1988/11	6	442	40.3	21.6	237
1988/10	8	477	37.5	19.4	247
1988/09	8	418	43.7	24.8	237
1988/08	3	403	50.1	25.1	202
1988/07	4	407	41.3	21.6	213
1988/06	5	384	59.6	27.2	175
1988/05	6	486	42.8	21.9	249
1988/04	10	382	50.4	27.2	206
1988/03	2	437	55.6	24.4	192
1988/02	3	419	45.9	24.1	220
1988/01	3	435	37.0	21.2	249
1987/12	4	350	37.6	18.7	174

1987/11	3	375	51.3	23.9	175
1987/10	1	359	58.3	26.5	163
1987/09	10	377	54.7	25.5	176
1987/08	3	365	43.9	25.9	215
1987/07	7	392	41.4	23.1	219
1987/06	3	353	55.4	28.9	184
1987/05	1	371	43.1	25.5	219
1987/04	2	363	49.6	26.6	195
1987/03	4	388	42.5	24.5	223
1987/02	4	391	50.6	23.4	181
1987/01	2	352	42.5	23.1	191

D. MH-FAILS, MH-ML1(2)-S, MH-ML1(2)-U DATA

TABLE 4. MAN-HOURS FOR FAILURE AND MAINTENANCE ACTION

MH-FAILS/MH-ML1-S/MH-ML1-U/MH-ML2-S/MH-ML2-U AFWA F-14A 74A1500 - T1224/AWG9 RADAR TRANSMITTER 0					
Date	MH-FAILS	MH-ML1-S	MH-ML1-U	MH-ML2-S	MH-ML2U
1991/12	806.6	0.6	1232.9	0.0	398.3
1991/11	1952.9	0.0	2420.1	0.0	1121.4
1991/10	2162.6	0.0	2084.7	0.0	1425.3
1991/09	2642.5	0.0	2310.7	0.0	1566.1
1991/08	3139.5	0.0	2760.5	0.0	2007.6
1991/07	2227.9	6.0	1915.8	6.1	1575.6
1991/06	2470.8	0.0	2310.4	0.0	1635.8
1991/05	2205.3	24.4	2317.2	14.6	1401.5
1991/04	2994.3	0.0	2522.0	0.0	1892.8
1991/03	3170.5	0.0	2690.0	0.0	1964.9
1991/02	3953.6	0.0	3028.9	0.0	2657.2
1991/01	4684.1	0.0	3586.9	0.0	3249.3
1990/12	2497.4	13.8	2210.5	3.5	1692.2

1990/11	2871.1	0.0	2576.0	0.0	1869.1
1990/10	2616.4	0.0	2458.1	0.0	1695.5
1990/09	2753.4	0.0	2454.1	0.0	1724.3
1990/08	3603.1	0.0	2920.7	0.0	2570.8
1990/07	2734.1	0.0	2461.4	0.0	1966.6
1990/06	2571.3	6.0	2107.8	0.0	1838.1
1990/05	2505.1	2.4	2416.9	0.0	1897.0
1990/04	2389.0	0.0	2344.9	0.0	1706.6
1990/03	3433.5	0.0	3017.9	0.0	2066.1
1990/02	2653.0	0.0	2795.8	0.0	1812.9
1990/01	3405.1	0.0	3804.8	0.0	2154.9
1989/12	1861.9	0.0	1841.5	0.0	1125.5
1989/11	2172.6	0.0	2577.6	0.0	1282.5
1989/10	3137.0	11.2	3221.0	6.2	2203.6
1989/09	2705.6	3.0	2965.1	0.0	1704.9
1989/08	3302.6	4.8	3318.3	4.8	2006.0
1989/07	2437.4	0.0	2505.7	0.0	1497.7
1989/06	2906.1	0.4	2572.7	0.0	1688.8
1989/05	2881.9	0.0	2447.1	0.0	1727.8
1989/04	3394.6	0.0	2895.9	0.0	2255.0
1989/03	2952.9	0.0	2739.5	0.0	2015.4
1989/02	2550.0	5.2	2526.6	0.0	1431.8
1989/01	3069.8	12.6	2582.5	0.0	1697.3
1988/12	2208.3	11.1	2387.2	0.0	1096.5
1988/11	2961.8	0.0	2695.2	0.0	1723.3
1988/10	3193.2	0.0	3634.9	0.0	2112.7
1988/09	3009.2	0.0	2620.1	0.0	1578.0
1988/08	2817.8	12.8	2404.0	0.0	1717.2
1988/07	2632.7	0.0	2162.2	0.0	1645.9
1988/06	2192.8	0.0	2216.8	0.0	1503.5

1988/05	3630.0	0.0	3099.4	0.0	2446.7
1988/04	2689.5	7.2	2394.8	4.2	1596.9
1988/03	2332.1	7.0	2626.1	0.0	1469.5
1988/02	3449.0	3.5	3207.1	0.0	1622.3
1988/01	3060.8	16.6	2746.8	8.0	1570.6
1987/12	2142.2	4.6	1872.9	18.7	1481.5
1987/11	2286.3	0.0	2242.9	0.0	1464.0
1987/10	2247.7	0.0	1942.0	0.0	1920.1
1987/09	2433.2	0.0	2106.9	0.0	1878.9
1987/08	3194.1	0.0	2449.1	0.0	1911.6
1987/07	3559.5	0.0	2964.9	0.0	1988.4
1987/06	2647.4	0.0	2210.4	0.0	1780.3
1987/05	2933.9	0.0	2166.4	0.0	1892.4
1987/04	2854.5	0.0	2551.3	0.0	1732.2
1987/03	2928.5	0.0	2219.5	0.0	1922.6
1987/02	2598.8	17.7	2336.6	3.8	1739.0
1987/01	2695.5	0.0	1832.6	0.0	1856.2

The "O" level MH/MA are shown by its components MH-ML1-S and MH-ML1-U. These must be added together on a month by month basis to get the total "O" level MH/MA. Similarly the "I" level MH/MA is made up of MH-ML2-S and MH-ML2-U.

APPENDIX G

PROGRAM OUTPUT

A. OUTPUT FOR SIMULATED DATA

1. Simulated data

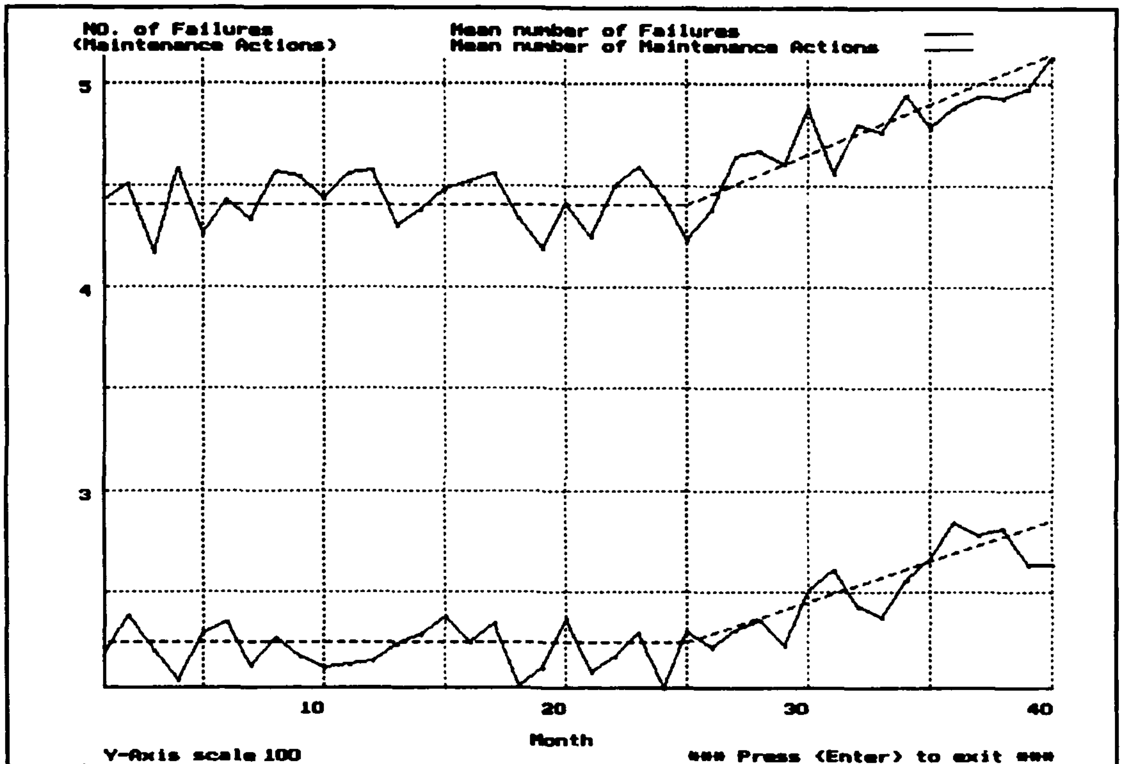

Figure 2. Simulated data.

"Mean number of" Failures	"Mean number of" Maintenance Actions
219.403	442.669
238.545	450.679
221.243	417.159
206.230	458.454
230.087	426.491
235.508	442.404
213.537	433.044
227.863	456.907

217.881	454.586
212.888	443.307
214.141	455.779
215.757	457.750
224.760	430.095
229.202	437.908
237.583	447.812
225.537	451.614
234.781	456.032
203.318	433.921
213.121	418.431
236.520	441.045
210.473	424.941
217.882	449.765
230.175	458.645
202.660	443.813
230.243	423.049
223.042	437.962
231.463	464.286
236.364	466.401
223.909	460.691
250.497	487.308
261.193	456.121
243.163	480.219
237.676	475.792
256.561	494.986
266.044	478.079
284.469	488.819
278.240	494.071
280.863	492.733
263.594	497.664
262.996	513.294

2. Bayesian procedure

```
Data (Read file: 0, Simulate: 1)          :          1
Simu. para. (MuF, EtaF, Sig2F, CF )       :  225     4   144    25
Simu. para. (MuMA,EtaMA,Sig2MA,CMA)       :  440     5   225    25
Reading filename                          :b:\f15.prn
NO. of systems & Use/Mon (NS.UPM)         :     400.00      25.00
New MTBF & MTBMA (MTBFNew,MTBMANew)       :      80.00      20.60
O MH/MA     old & new (OMHOld,OMHNew)     :       6.34       9.01
I MH/MA     old & new (IMHOLd,IMHNew)     :       9.01       9.01
MH/F        old & new (FMHOld,FMHNew)     :      13.13      13.13
MATL Cost    O & N (MATLOld,MATLNew)      :    1082.64    1082.64
NO. of BCMs O & N (BCMOld,BCMNew)         :       7.80       4.00
AVDLR Cost   O & N (AVDLROld,AVDLRNew)    :    1120.36    1120.36
Fix cost (CF)                             :    5400000
Lead Time & Kits/Mo (LeadTime,INSTL)      :         60         15
Time start & horizon (ST,Hor)             :          6        150
```

Method (Likelihood: 0, Bayesian: 1) : 1
Replication (Rep) : 100

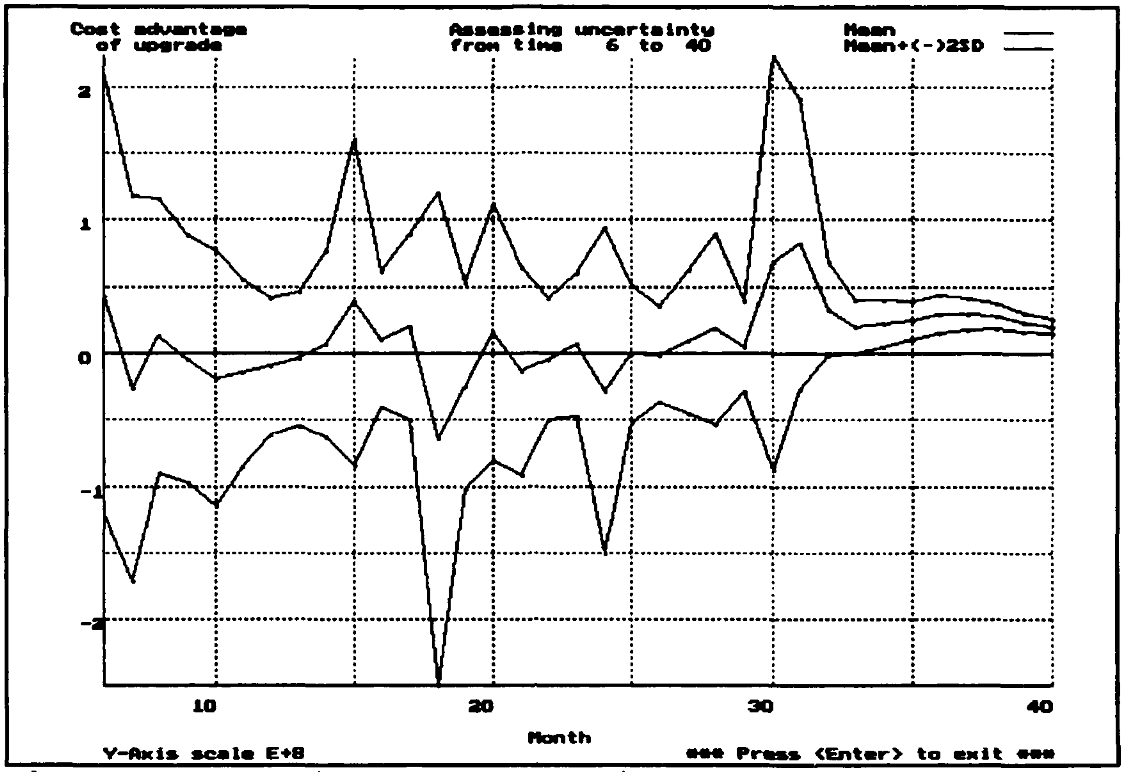

Figure 3. Bayesian result for simulated data.

Bayesian procedure :
The best time for subsystem upgrade
and assessing uncertainty
(from time 6 to time 40)

"Time" Index	"Best " upgrade time	" Cost " advantage of upgrade	"Std.Dev." of cost	"Mean-2SD" bound	"Mean+2SD" bound
6	6	4.546E+07	8.266E+07	-1.199E+08	2.108E+08
7	150	-2.614E+07	7.234E+07	-1.708E+08	1.185E+08
8	8	1.305E+07	5.148E+07	-8.992E+07	1.160E+08
9	150	-4.607E+06	4.636E+07	-9.734E+07	8.812E+07
10	150	-1.937E+07	4.778E+07	-1.149E+08	7.619E+07
11	150	-1.434E+07	3.520E+07	-8.475E+07	5.607E+07
12	150	-9.970E+06	2.588E+07	-6.173E+07	4.179E+07
13	150	-4.122E+06	2.527E+07	-5.467E+07	4.642E+07
14	14	7.006E+06	3.476E+07	-6.250E+07	7.652E+07
15	15	3.878E+07	6.105E+07	-8.333E+07	1.609E+08
16	16	1.056E+07	2.539E+07	-4.023E+07	6.134E+07
17	17	1.968E+07	3.486E+07	-5.005E+07	8.940E+07
18	150	-6.433E+07	9.237E+07	-2.491E+08	1.204E+08

19	150	-2.441E+07	3.863E+07	-1.017E+08	5.285E+07
20	20	1.567E+07	4.822E+07	-8.078E+07	1.121E+08
21	150	-1.315E+07	3.870E+07	-9.054E+07	6.425E+07
22	150	-4.170E+06	2.271E+07	-4.959E+07	4.125E+07
23	23	6.650E+06	2.699E+07	-4.734E+07	6.064E+07
24	150	-2.807E+07	6.101E+07	-1.501E+08	9.395E+07
25	150	-1.630E+05	2.570E+07	-5.156E+07	5.124E+07
26	150	-6.507E+05	1.788E+07	-3.642E+07	3.511E+07
27	27	9.095E+06	2.684E+07	-4.459E+07	6.278E+07
28	28	1.845E+07	3.560E+07	-5.274E+07	8.964E+07
29	29	5.109E+06	1.667E+07	-2.822E+07	3.844E+07
30	30	6.800E+07	7.760E+07	-8.721E+07	2.232E+08
31	31	8.249E+07	5.462E+07	-2.676E+07	1.917E+08
32	32	3.344E+07	1.722E+07	-1.006E+06	6.789E+07
33	33	2.004E+07	1.003E+07	-1.352E+04	4.010E+07
34	34	2.272E+07	8.962E+06	4.799E+06	4.065E+07
35	35	2.468E+07	7.023E+06	1.063E+07	3.872E+07
36	36	2.960E+07	7.079E+06	1.544E+07	4.375E+07
37	37	2.929E+07	5.986E+06	1.732E+07	4.126E+07
38	38	2.815E+07	4.898E+06	1.835E+07	3.794E+07
39	39	2.365E+07	3.521E+06	1.661E+07	3.070E+07
40	40	2.064E+07	2.871E+06	1.490E+07	2.638E+07

3. Maximum likelihood procedure

Data (Read file: 0, Simulate: 1)	:		0		
Simu. para. (MuF, EtaF, Sig2F, CF)	:	0	0	0	0
Simu. para. (MuMA,EtaMA,Sig2MA,CMA)	:	0	0	0	0
Reading filename	:b:\Simu.Data				
NO. of systems & Use/Mon (NS.UPM)	:	400.00		25.00	
New MTBF & MTBMA (MTBFNew,MTBMANew)	:	80.00		20.60	
O MH/MA old & new (OMHOld,OMHNew)	:	6.34		9.01	
I MH/MA old & new (IMHOLd,IMHNew)	:	9.01		9.01	
MH/F old & new (FMHOld,FMHNew)	:	13.13		13.13	
MATL Cost O & N (MATLOld,MATLNew)	:	1082.64		1082.64	
NO. of BCMs O & N (BCMOld,BCMNew)	:	7.80		4.00	
AVDLR Cost O & N (AVDLROld,AVDLRNew)	:	1120.36		1120.36	
Fix cost (CF)	:	5400000			
Lead Time & Kits/Mo (LeadTime,INSTL)	:	60		15	
Time start & horizon (ST,Hor)	:	6		150	
Method (Likelihood: 0, Bayesian: 1)	:	0			
Replication (Rep)	:	100			

Maximum likelihood procedure :
The best time for subsystem upgrade
and assessing uncertainty
(from time 6 to time 40)

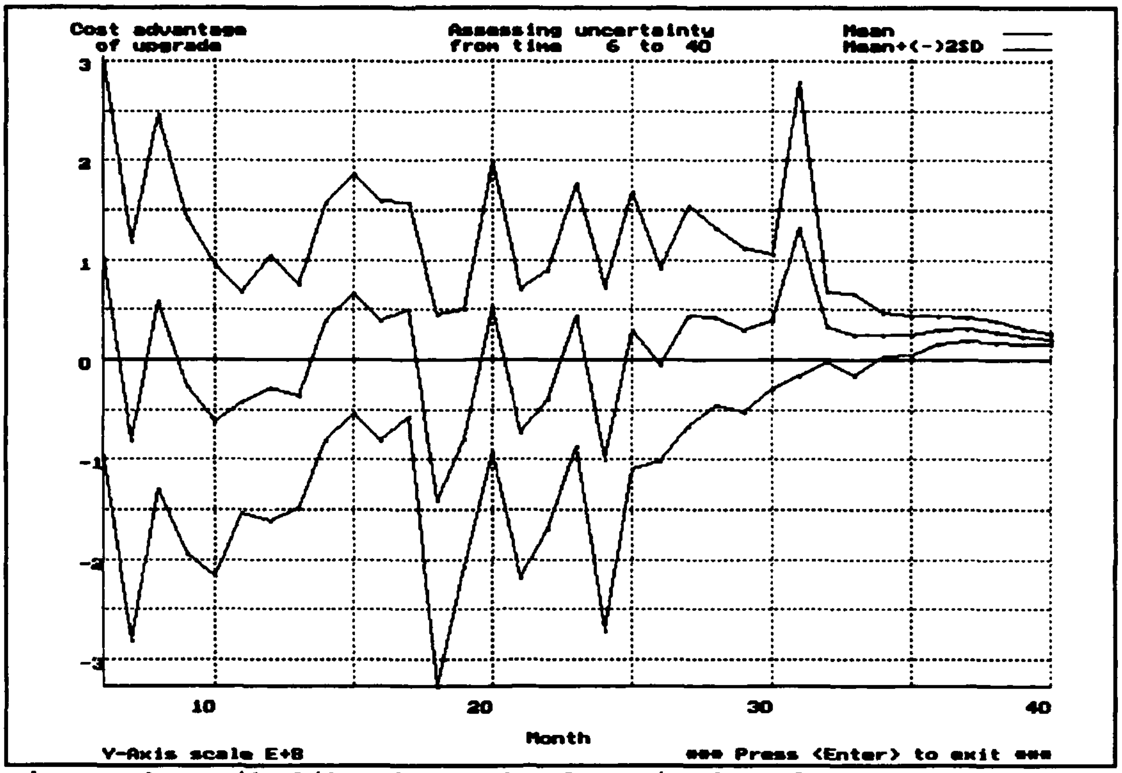

Figure 4. Likelihood result for simulated data.

"Time" Index	"Best " upgrade time	" Cost " advantage of upgrade	"Std.Dev." of cost	"Mean-2SD" bound	"Mean+2SD" bound
6	6	1.073E+08	9.894E+07	-9.059E+07	3.052E+08
7	150	-8.105E+07	9.998E+07	-2.810E+08	1.189E+08
8	8	5.838E+07	9.401E+07	-1.297E+08	2.464E+08
9	150	-2.674E+07	8.415E+07	-1.950E+08	1.416E+08
10	150	-6.119E+07	7.800E+07	-2.172E+08	9.481E+07
11	150	-4.226E+07	5.527E+07	-1.528E+08	6.828E+07
12	150	-2.887E+07	6.639E+07	-1.616E+08	1.039E+08
13	150	-3.698E+07	5.617E+07	-1.493E+08	7.536E+07
14	14	3.907E+07	5.981E+07	-8.054E+07	1.587E+08
15	15	6.615E+07	5.991E+07	-5.366E+07	1.860E+08
16	16	3.937E+07	6.020E+07	-8.102E+07	1.598E+08
17	17	5.077E+07	5.354E+07	-5.630E+07	1.579E+08
18	150	-1.407E+08	9.320E+07	-3.271E+08	4.573E+07
19	150	-7.840E+07	6.436E+07	-2.071E+08	5.032E+07
20	20	5.281E+07	7.342E+07	-9.404E+07	1.997E+08
21	150	-7.279E+07	7.229E+07	-2.174E+08	7.180E+07
22	150	-3.974E+07	6.503E+07	-1.698E+08	9.032E+07
23	23	4.448E+07	6.606E+07	-8.764E+07	1.766E+08
24	150	-9.913E+07	8.607E+07	-2.713E+08	7.301E+07
25	25	2.967E+07	6.944E+07	-1.092E+08	1.686E+08
26	150	-4.887E+06	4.829E+07	-1.015E+08	9.170E+07

27	27	4.409E+07	5.510E+07	-6.610E+07	1.543E+08
28	28	4.342E+07	4.466E+07	-4.589E+07	1.327E+08
29	29	3.031E+07	4.090E+07	-5.149E+07	1.121E+08
30	30	3.892E+07	3.378E+07	-2.863E+07	1.065E+08
31	31	1.323E+08	7.389E+07	-1.547E+07	2.801E+08
32	32	3.311E+07	1.720E+07	-1.287E+06	6.751E+07
33	33	2.562E+07	2.071E+07	-1.579E+07	6.704E+07
34	34	2.486E+07	1.091E+07	3.027E+06	4.668E+07
35	35	2.458E+07	9.901E+06	4.773E+06	4.438E+07
36	36	3.044E+07	6.968E+06	1.650E+07	4.438E+07
37	37	3.097E+07	5.708E+06	1.956E+07	4.239E+07
38	38	2.867E+07	5.358E+06	1.795E+07	3.938E+07
39	39	2.427E+07	3.867E+06	1.653E+07	3.200E+07
40	40	2.113E+07	2.942E+06	1.524E+07	2.701E+07

B. OUTPUT FOR REAL DATA

1. Real data

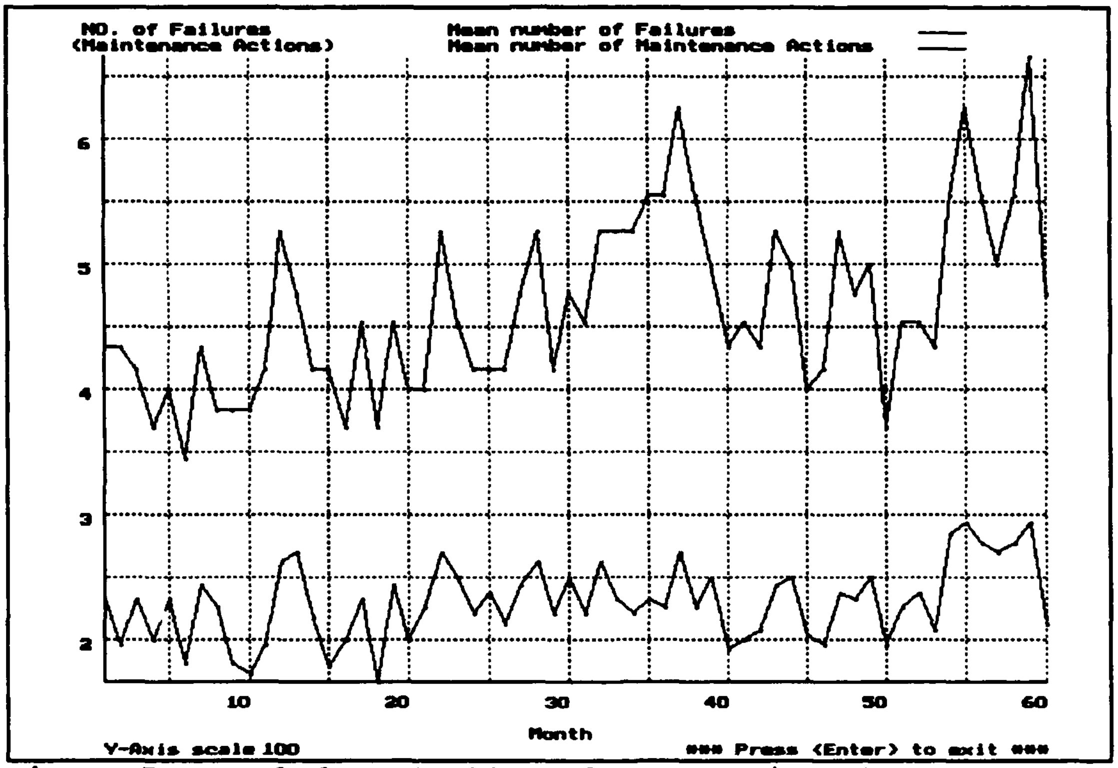

Figure 5. Real data (F-14A radar transmitter).

"Mean number of" Failures   "Mean number of" Maintenance Actions

232.558	434.783
196.078	434.783
232.558	416.667
200.000	370.370
232.558	400.000
181.818	344.828
243.902	434.783
227.273	384.615
181.818	384.615
172.414	384.615
196.078	416.667
263.158	526.316
270.270	476.190
217.391	416.667
178.571	416.667
200.000	370.370
232.558	454.545
166.667	370.370
243.902	454.545
200.000	400.000
227.273	400.000
270.270	526.316
250.000	454.545
222.222	416.667
238.095	416.667
212.766	416.667
243.902	476.190
263.158	526.316
222.222	416.667
250.000	476.190
222.222	454.545
263.158	526.316
232.558	526.316
222.222	526.316
232.558	555.556
227.273	555.556
270.270	625.000
227.273	555.556
250.000	500.000
192.308	434.783
200.000	454.545
208.333	434.783
243.902	526.316
250.000	500.000
204.082	400.000
196.078	416.667
238.095	526.316
232.558	476.190
250.000	500.000
196.078	370.370
227.273	454.545

238.095	454.545
208.333	434.783
285.714	555.556
294.118	625.000
277.778	555.556
270.270	500.000
277.778	555.556
294.118	666.667
212.766	476.190

2. Bayesian procedure

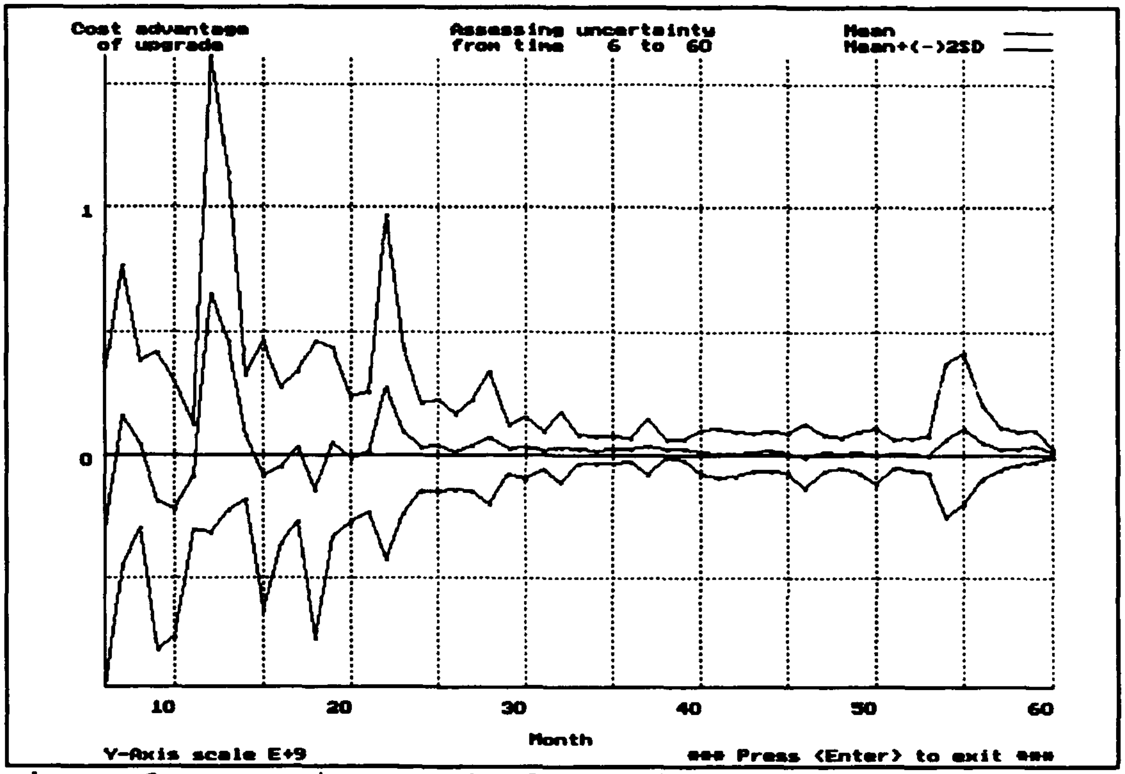

Figure 6. Bayesian result for real data.

Data (Read file: 0, Simulate: 1)	:		0		
Simu. para. (MuF, EtaF, Sig2F, CF)	:	0	0	0	0
Simu. para. (MuMA,EtaMA,Sig2MA,CMA)	:	0	0	0	0
Reading filename	:b:\f15.prn				
NO. of systems & Use/Mon (NS.UPM)	:		400.00		25.00
New MTBF & MTBMA (MTBFNew,MTBMANew)	:		80.00		20.60
O MH/MA old & new (OMHOld,OMHNew)	:		6.34		9.01
I MH/MA old & new (IMHOLd,IMHNew)	:		9.01		9.01
MH/F old & new (FMHOld,FMHNew)	:		13.13		13.13
MATL Cost O & N (MATLOld,MATLNew)	:		1082.64		1082.64
NO. of BCMs O & N (BCMOld,BCMNew)	:		7.80		4.00

AVDLR Cost O & N (AVDLROld,AVDLRNew)	:	1120.36	1120.36
Fix cost (CF)	:	5400000	
Lead Time & Kits/Mo (LeadTime,INSTL)	:	60	15
Time start & horizon (ST,Hor)	:	6	180
Method (Likelihood: 0, Bayesian: 1)	:	1	
Replication (Rep)	:	100	

Bayesian procedure :
The best time for subsystem upgrade
and assessing uncertainty
(from time 6 to time 60)

"Time" Index	"Best " upgrade time	" Cost " advantage of upgrade	"Std.Dev." of cost	"Mean-2SD" bound	"Mean+2SD" bound
6	180	-3.031E+08	3.154E+08	-9.338E+08	3.276E+08
7	7	1.583E+08	3.020E+08	-4.456E+08	7.622E+08
8	8	4.605E+07	1.692E+08	-2.923E+08	3.844E+08
9	180	-1.832E+08	2.988E+08	-7.808E+08	4.145E+08
10	180	-2.162E+08	2.571E+08	-7.304E+08	2.980E+08
11	180	-8.537E+07	1.079E+08	-3.013E+08	1.305E+08
12	12	6.514E+08	4.816E+08	-3.118E+08	1.615E+09
13	13	4.591E+08	3.398E+08	-2.206E+08	1.139E+09
14	14	7.408E+07	1.270E+08	-1.798E+08	3.280E+08
15	180	-8.311E+07	2.696E+08	-6.223E+08	4.561E+08
16	180	-4.158E+07	1.589E+08	-3.595E+08	2.763E+08
17	17	3.493E+07	1.510E+08	-2.670E+08	3.369E+08
18	180	-1.397E+08	3.003E+08	-7.404E+08	4.610E+08
19	19	5.106E+07	1.915E+08	-3.320E+08	4.341E+08
20	180	-1.503E+07	1.259E+08	-2.668E+08	2.368E+08
21	21	1.374E+07	1.218E+08	-2.299E+08	2.574E+08
22	22	2.720E+08	3.470E+08	-4.220E+08	9.660E+08
23	23	9.428E+07	1.649E+08	-2.355E+08	4.240E+08
24	24	3.181E+07	8.776E+07	-1.437E+08	2.073E+08
25	25	3.559E+07	9.217E+07	-1.487E+08	2.199E+08
26	26	1.502E+07	7.652E+07	-1.380E+08	1.681E+08
27	27	3.652E+07	9.307E+07	-1.496E+08	2.227E+08
28	28	7.122E+07	1.333E+08	-1.953E+08	3.378E+08
29	29	2.423E+07	4.937E+07	-7.450E+07	1.230E+08
30	30	3.123E+07	6.212E+07	-9.301E+07	1.555E+08
31	31	2.130E+07	3.823E+07	-5.516E+07	9.776E+07
32	32	3.330E+07	6.929E+07	-1.053E+08	1.719E+08
33	33	2.517E+07	2.952E+07	-3.388E+07	8.421E+07
34	34	2.187E+07	2.656E+07	-3.126E+07	7.499E+07
35	35	2.335E+07	2.654E+07	-2.973E+07	7.643E+07
36	36	2.294E+07	2.492E+07	-2.691E+07	7.279E+07
37	37	3.770E+07	5.581E+07	-7.392E+07	1.493E+08
38	38	2.821E+07	1.798E+07	-7.757E+06	6.417E+07
39	39	2.407E+07	2.013E+07	-1.620E+07	6.433E+07
40	40	1.301E+07	4.390E+07	-7.479E+07	1.008E+08
41	41	7.387E+06	4.962E+07	-9.186E+07	1.066E+08

42	42	6.332E+06	4.417E+07	-8.201E+07	9.467E+07
43	43	1.468E+07	3.849E+07	-6.231E+07	9.167E+07
44	44	1.615E+07	3.932E+07	-6.249E+07	9.479E+07
45	45	6.863E+06	3.979E+07	-7.272E+07	8.645E+07
46	180	-3.456E+06	6.481E+07	-1.331E+08	1.262E+08
47	47	1.015E+07	3.697E+07	-6.379E+07	8.409E+07
48	48	9.418E+06	3.168E+07	-5.394E+07	7.278E+07
49	49	1.316E+07	4.153E+07	-6.990E+07	9.623E+07
50	180	-1.373E+06	5.540E+07	-1.122E+08	1.094E+08
51	51	5.869E+06	2.976E+07	-5.365E+07	6.539E+07
52	52	8.059E+06	3.121E+07	-5.437E+07	7.049E+07
53	53	1.619E+06	3.606E+07	-7.051E+07	7.375E+07
54	54	6.219E+07	1.549E+08	-2.475E+08	3.719E+08
55	55	1.092E+08	1.535E+08	-1.979E+08	4.163E+08
56	56	5.206E+07	7.508E+07	-9.810E+07	2.022E+08
57	57	2.849E+07	4.166E+07	-5.483E+07	1.118E+08
58	58	2.637E+07	3.311E+07	-3.986E+07	9.260E+07
59	59	3.835E+07	3.285E+07	-2.735E+07	1.040E+08
60	60	9.829E+06	7.842E+06	-5.855E+06	2.551E+07

3. Maximum likelihood procedure

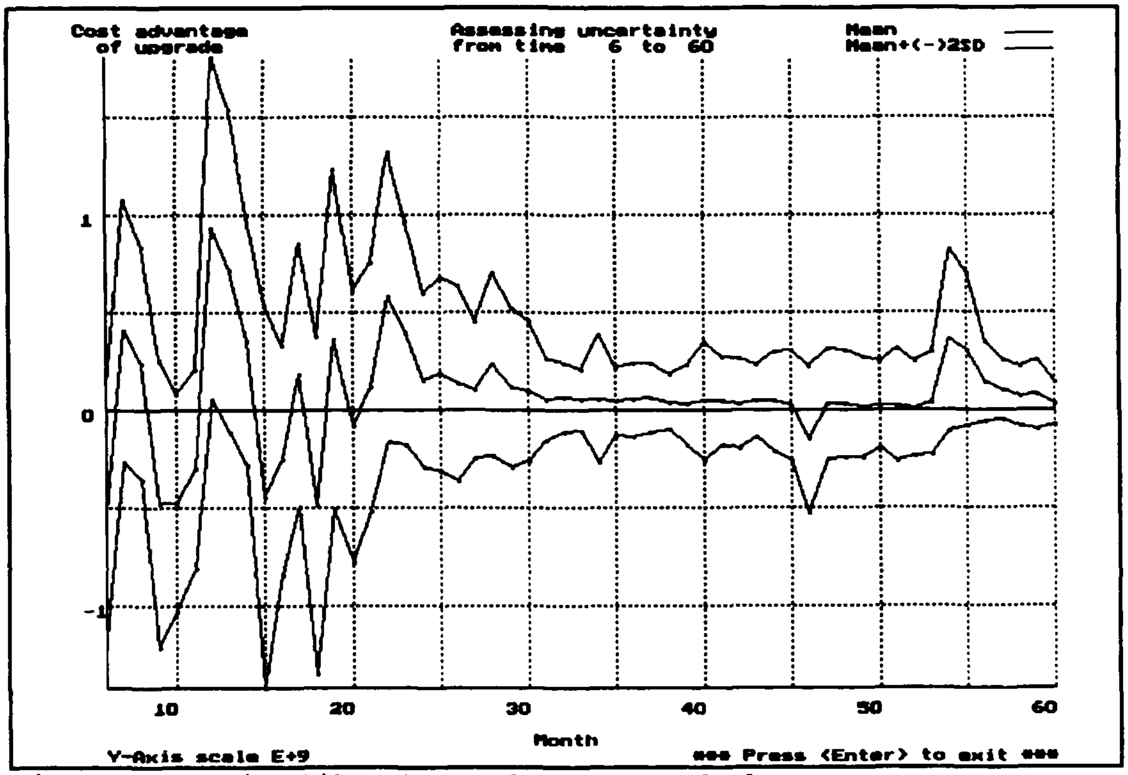

Figure 7. Likelihood result for real data.

Data (Read file: 0, Simulate: 1)	:	0			
Simu. para. (MuF, EtaF, Sig2F, CF)	:	0	0	0	0
Simu. para. (MuMA,EtaMA,Sig2MA,CMA)	:	0	0	0	0
Reading filename	:b:\f15.prn				
NO. of systems & Use/Mon (NS.UPM)	:	400.00		25.00	
New MTBF & MTBMA (MTBFNew,MTBMANew)	:	80.00		20.60	
O MH/MA old & new (OMHOld,OMHNew)	:	6.34		9.01	
I MH/MA old & new (IMHOLd,IMHNew)	:	9.01		9.01	
MH/F old & new (FMHOld,FMHNew)	:	13.13		13.13	
MATL Cost O & N (MATLOld,MATLNew)	:	1082.64		1082.64	
NO. of BCMs O & N (BCMOld,BCMNew)	:	7.80		4.00	
AVDLR Cost O & N (AVDLROld,AVDLRNew)	:	1120.36		1120.36	
Fix cost (CF)	:	5400000			
Lead Time & Kits/Mo (LeadTime,INSTL)	:	60		15	
Time start & horizon (ST,Hor)	:	6		180	
Method (Likelihood: 0, Bayesian: 1)	:	0			
Replication (Rep)	:	100			

Maximum likelihood procedure :
The best time for subsystem upgrade
and assessing uncertainty
(from time 6 to time 60)

"Time" Index	"Best " upgrade time	" Cost " advantage of upgrade	"Std.Dev." of cost	"Mean-2SD" bound	"Mean+2SD" bound
6	180	-5.224E+08	3.242E+08	-1.171E+09	1.260E+08
7	7	4.100E+08	3.364E+08	-2.628E+08	1.083E+09
8	8	2.383E+08	2.984E+08	-3.585E+08	8.352E+08
9	180	-4.743E+08	3.667E+08	-1.208E+09	2.591E+08
10	180	-4.726E+08	2.750E+08	-1.023E+09	7.748E+07
11	180	-3.022E+08	2.540E+08	-8.102E+08	2.058E+08
12	12	9.329E+08	4.386E+08	5.578E+07	1.810E+09
13	13	7.144E+08	4.120E+08	-1.097E+08	1.538E+09
14	14	3.542E+08	3.189E+08	-2.836E+08	9.919E+08
15	180	-4.399E+08	4.887E+08	-1.417E+09	5.376E+08
16	180	-2.532E+08	2.931E+08	-8.394E+08	3.330E+08
17	17	1.832E+08	3.368E+08	-4.905E+08	8.569E+08
18	180	-4.825E+08	4.317E+08	-1.346E+09	3.808E+08
19	19	3.658E+08	4.353E+08	-5.049E+08	1.236E+09
20	180	-8.019E+07	3.485E+08	-7.772E+08	6.168E+08
21	21	1.228E+08	3.198E+08	-5.167E+08	7.624E+08
22	22	5.815E+08	3.718E+08	-1.621E+08	1.325E+09
23	23	3.945E+08	2.859E+08	-1.773E+08	9.664E+08
24	24	1.514E+08	2.211E+08	-2.908E+08	5.936E+08
25	25	1.825E+08	2.479E+08	-3.133E+08	6.782E+08
26	26	1.375E+08	2.518E+08	-3.662E+08	6.411E+08
27	27	1.070E+08	1.742E+08	-2.414E+08	4.555E+08
28	28	2.338E+08	2.335E+08	-2.333E+08	7.009E+08
29	29	1.203E+08	2.037E+08	-2.870E+08	5.277E+08
30	30	9.947E+07	1.792E+08	-2.588E+08	4.578E+08

31	31	5.994E+07	1.046E+08	-1.494E+08	2.692E+08
32	32	6.364E+07	8.879E+07	-1.139E+08	2.412E+08
33	33	5.474E+07	7.949E+07	-1.042E+08	2.137E+08
34	34	5.988E+07	1.618E+08	-2.638E+08	3.835E+08
35	35	4.438E+07	8.703E+07	-1.297E+08	2.184E+08
36	36	5.332E+07	9.323E+07	-1.331E+08	2.398E+08
37	37	6.361E+07	9.020E+07	-1.168E+08	2.440E+08
38	38	4.272E+07	7.133E+07	-9.994E+07	1.854E+08
39	39	2.866E+07	1.017E+08	-1.748E+08	2.321E+08
40	40	4.781E+07	1.511E+08	-2.544E+08	3.500E+08
41	41	5.147E+07	1.126E+08	-1.736E+08	2.766E+08
42	42	3.562E+07	1.164E+08	-1.972E+08	2.684E+08
43	43	4.570E+07	9.229E+07	-1.389E+08	2.303E+08
44	44	4.500E+07	1.263E+08	-2.076E+08	2.976E+08
45	45	2.252E+07	1.419E+08	-2.612E+08	3.062E+08
46	180	-1.480E+08	1.886E+08	-5.251E+08	2.291E+08
47	47	3.083E+07	1.419E+08	-2.531E+08	3.147E+08
48	48	3.112E+07	1.360E+08	-2.408E+08	3.031E+08
49	49	1.914E+07	1.288E+08	-2.385E+08	2.768E+08
50	50	2.607E+07	1.104E+08	-1.947E+08	2.468E+08
51	51	2.750E+07	1.423E+08	-2.571E+08	3.121E+08
52	52	7.305E+06	1.204E+08	-2.336E+08	2.482E+08
53	53	3.637E+07	1.318E+08	-2.272E+08	3.000E+08
54	54	3.591E+08	2.313E+08	-1.034E+08	8.217E+08
55	55	3.088E+08	1.963E+08	-8.384E+07	7.015E+08
56	56	1.464E+08	9.983E+07	-5.321E+07	3.461E+08
57	57	1.027E+08	7.619E+07	-4.965E+07	2.551E+08
58	58	7.427E+07	7.684E+07	-7.942E+07	2.280E+08
59	59	7.846E+07	8.815E+07	-9.784E+07	2.548E+08
60	60	3.585E+07	5.340E+07	-7.094E+07	1.426E+08

APPENDIX H

UPGRADE.PAS PROGRAM

A. USER DOCUMENTATION

This documentation contains the information concerning the utilization, input data, and results of the UPGRADE.PAS program.

1. Utilization

This decision aid program (UPGRADE.PAS) is developed to enhance the Economic Analysis program (ROI) in the Automated Management Indicator System (AMIS). It uses statistical procedures to estimate the time of onset of subsystem degradation and the magnitude and evolution of the degradation over time. These estimates are used to compute the estimated cost of remaining with the current subsystem. A comparison of this cost with the cost of investing in the upgraded subsystem can suggest to the user a most economical time to upgrade this particular subsystem. The analysis includes assessment of the uncertainty in the estimated costs.

This program is based on the model, which postulates that there may be a linear trend in the mean number of failures (maintenance actions) per time period for the current subsystem. It uses a statistical method to detect and quantify

the trend and combines these with the other values to estimate the future costs for the subsystem.

The program is written in TURBO PASCAL for use on a personal computer, not for the mainframe. It also includes screen graphics. The default screen graphics is VGA. The user should adjust the value of "GraphDriver", "GraphMode", and the path of "InitGraph" in procedures "GenDataGraph" and "GenGraph" before using the program. The graphical output cannot be printed by pressing <Print Screen> on keyboard. The user needs other software (Colorix etc.) to get a hard copy of the graphs.

2. Menu and Input Data

When the program is run, the menu will appear on the screen and wait for further adjustments to default values of the parameters. Permanent changes to parameter values should be done in the procedure "DefaultParameter" of the source code. All input data was chosen to be similar to those used in the ROI program, so the values of the parameters can be obtained from other existing data bases (NALDA etc.). The main difference between the UPGRADE.PAS and ROI programs is the estimation of the mean number of failures (respectively maintenance actions) for the current subsystem. In UPGRADE.PAS we use the whole collected data array to detect and quantify the trend. The resulting estimates are used to estimate the

future costs. The ROI program simply uses the average values of the measures of performance for the past 24 months data.

A listing of the menu, excluding default values for the parameters, is

```
 0  Run the model.
 1  Data (Read file: 0, Simulate: 1)         :
 2  Simu. para. (MeuF, EtaF, Sig2F, CF )     :
 3  Simu. para. (MeuMA,EtaMA,Sig2MA,CMA)     :
 4  Reading filename                         :
 5  NO. of systems & Use/Mon (NS.UPM)        :
 6  New MTBF & MTBMA (MTBFNew,MTBMANew)      :
 7  O MH/MA    old & new (OMHOld,OMHNew)     :
 8  I MH/MA    old & new (IMHOLd,IMHNew)     :
 9  MH/F       old & new (FMHOld,FMHNew)     :
10  MATL Cost   O & N (MATLOld,MATLNew)      :
11  NO. of BCMs O & N (BCMOld,BCMNew)        :
12  AVDLR Cost  O & N (AVDLROld,AVDLRNew)    :
13  Fix Cost (CF)                            :
14  Lead time & Kits/Mo (LeadTime,INSTL)     :
15  Time start & horizon (ST,Hor)            :
16  Method (Likelihood: 0, Bayesian: 1)      :
17  Replication (Rep)                        :
18  Reset to default values.
```

Items 1 through 4 ask the user to choose the data array of mean number of failures (respectively maintenance actions) for the current subsystem (item 1). If the user

elects to simulate these two arrays, then he/she should input the values of the four model parameters for each array; these are the mean value before the trend, the slope of the trend, the standard deviation of the data array, and the occurrence time of the trend (items 2 and 3). Item 2 requests the parameters used for the number of failures for the simulation; item 3 requests the parameters used for the number of maintenance actions for the simulation. If the user decides to analyze outside data, a ASCII file is needed containing the MFHBF and MFHBMA arrays in the NALDA data base (item 4). The format of the file should have two columns. The first column is MFHBF; the second column is MFHBMA (see columns 4 and 5 in Table 3). The UPGRADE.PAS program will transform these two arrays to the mean number of failures and maintenance actions.

Items 5 through 12 can be obtained from the ROI program in the "Current System" and "Improved System" sections. The user inputs the same values for the identified items. The UPGRADE.PAS program is designed to read the pairwise parameters (both systems) together. The parameters have the same heading followed by "Old" for the current system parameters; those followed by "New" are the improved system parameters.

Items 13 and 14 can also be obtained from the ROI program in the "Cost of Fix" and "Schedule for Fix" section. The items "Cost/Kit" and "Cost for INSTL/Kit" should be multiplied by number of systems (item 5) and added to the

other items in the cost of fix section to get "CF" (item 13). To compute "LeadTime", the user needs to add all the items together except the last item "Kits/MO INSTL" in the "Schedule for Fix" section. The value of "INSTL" is identical to that of "Kits/MO INSTL" in the ROI program (both at item 14).

Items 15 to 17 ask the user to choose the desired time in the time series to start the estimation of the best upgrading policy and mean cost advantage, and the expected time horizon for the subsystem (item 15). Then it provides two methods (the Maximum Likelihood and Bayesian) to estimate costs. The Bayesian method requires less computation than the Maximum Likelihood method (item 16). To only obtain the best policies for each time for the chosen procedure, set the parameter Rep=0. If Rep is a positive integer, then assessments of uncertainty for the best policy will also be given. If the Bayesian procedure is chosen, the assessment of uncertainty uses moments of the posterior distribution. If the maximum likelihood is used, Rep is equal to the number of replications for the bootstrap estimates of uncertainty (item 17). If the Bayesian procedure is chosen the time to start the calculation, ST=6.

The values of "MA/ML1(2) Ratio" and "ML1(2) Rate" can only be changed in the source code. They are stored in the constant declaration of the main program.

3. Results

When the program executes (choose item 0), three sections of results will appear. First, the graphs of the mean number of failures and mean number of maintenance actions will appear on the screen. Then, the most economical time for upgrading and estimated cost advantage of upgrade and two standard deviation bounds will appear after each time index (from the time chosen to start the calculation, ST, to the end of the time series, ET). Finally, the graphs of the mean cost advantage and two standard deviation bounds will appear on the screen. After looking the graph, user may press <Enter> to return the original screen.

The results, excluding the graphics, will be located in the F.OUT file. Simu.Data file contains the simulated data (mean time between failure and mean time between maintenance action) for the purpose of reuse.

B. SOURCE CODE

```
{$M 36384,0,655360}
PROGRAM UPGRADE;
Uses Dos, Crt, Graph;

{  This program is developed to estimate the time of onset of
a given (current) subsystem degradation and the magnitude of
the degradation. These estimates are then used to estimate the
cost of remaining with the current subsystem for the remaining
time horizon, the life of the parent system. We compare this
cost with the cost of investing in the upgraded (improved)
subsystem to obtain a best time to invest in the upgraded
subsystem.
    Three options are given in this program:
1.  Data acquisition is read from a data file (set Data = 0)
or simulated from the program itself by choosing parameters
```

```
(Data = 1).
2. Estimation uses the Maximum Likelihood procedure (set Meth
= 0) or the Bayesian procedure (Meth = 1).
3. Assess uncertainty or not. Set (Rep = 0) when the answer
is no. Otherwise the value of Rep represents the number of
bootstrap replication. The Bayesian procedure will
automatically assess uncertainty when Rep > 0 and Meth = 1.

    All input data should be copied from the ROI program
except the MFHBF and MFHBMA arrays. These arrays should be
contained in input file. They are obtained from the NALDA data
base.
    The values of "MA/ML1(2) Ratio" and "ML1(2) Rate" can only
be changed in the source code. They are stored in the constant
declaration of the main program. The other parameters can be
adjusted in the menu given on the computer screen.
    The length of "IntVec" and "RealVec" should be larger than
or equal to the value of "ET+1".
    The length of "RealVec1" should be larger than or equal to
the value of "Rep".

    When this program running, the graph of the mean number of
failures and mean number of maintenance actions will appear on
the screen. It is followed by the estimated best time for
upgrading the subsystem that will be printed after each
decision time index.
    If the user chooses to assess the uncertainty of
estimation, then the cost advantage of the best upgrade policy
and two standard deviation bounds will be printed on the
screen following the best upgrading time. The user can view
the graph and press <Enter> to leave graphical screen.

    The results, excluding the graphics, are located in F.OUT
(Output) file. This file also contains the desired parameters
(menu) for the computation.
    Simu.Data (Output2) contains the simulated data for the
purpose of reuse if Data = 1.                                }

const MAML1Ratio = 0.92;
      MAML2Ratio = 0.47;
      ML1Rate    = 15.28;
      ML2Rate    = 18.35;

type  IntVec    = array[1..121]  of integer;
      RealVec   = array[1..121]  of real;
      RealVec1  = array[1..1000] of real;
      StrVec    = array[0..18]  of string;

var   CTVec                      : IntVec;

      NOFVec, NOMAVec            : RealVec;
```

```
SimuFVec, SimuMAVec       : RealVec;
MuF, EtaF, SigmaSqrF      : RealVec;
MuMA, EtaMA, SigmaSqrMA   : RealVec;
VSqrF, RSqrF, RhoSqrF     : RealVec;
VSqrMA, RSqrMA, RhoSqrMA  : RealVec;
PiStarF, PiStarMA         : RealVec;
PiDistF, PiDistMA         : RealVec;
Mean, LB, UB              : RealVec;
Gain                      : RealVec1;

Mission                   : StrVec;

RanSeed                   : longint;
Data, Meth                : integer;
ST, ET, Hor               : integer;
LeadTime, INSTL           : integer;
Rep, R, Time, CT          : integer;
SetCF, SetCF1, HatCF      : integer;
SetCMA, SetCMA1, HatCMA   : integer;

MTBFNew, MTBMANew         : real;
NOFNew, NOMANew           : real;
OMHOld, OMHNew            : real;
IMHOld, IMHNew            : real;
FMHOld, FMHNew            : real;
MATLOld, MATLNew          : real;
BCMOld, BCMNew            : real;
AVDLROld, AVDLRNew        : real;
COMA, COF, COAD           : real;
CNMA, CNF, CNAD           : real;
CF, NS, UPM               : real;
SetMuF, SetEtaF           : real;
SetSigmaSqrF              : real;
SetMuF1, SetEtaF1         : real;
SetSigmaSqrF1             : real;
HatMuF, HatEtaF           : real;
HatSigmaSqrF              : real;
SetMuMA, SetEtaMA         : real;
SetSigmaSqrMA             : real;
SetMuMA1, SetEtaMA1       : real;
SetSigmaSqrMA1            : real;
HatMuMA, HatEtaMA         : real;
HatSigmaSqrMA             : real;
Max, Min, Total           : real;

Infile                    : string;

Output, Output2           : text;
Input                     : text;

{+++PART1 : SET MENU & INPUT DATA+++++++++++++++++++++++++}
```

```
{   These procedures set up the menu on the screen and input
values for the parameters. It includes:
    DefaultParameter,
    SetUnitCost,
    SetMission,
    SetParameter,
    WriteParameter.                                                }

Procedure DefaultParameter;

begin
   Data           := 0;
   SetMuF1        := 225;
   SetEtaF1       := 10;
   SetSigmaSqrF1  := 225;
   SetCF1         := 10;
   SetMuMA1       := 440;
   SetEtaMA1      := 10;
   SetSigmaSqrMA1 := 400;
   SetCMA1        := 10;

   NS             := 400;
   UPM            := 25;

   Infile         := 'a:\f15.prn';
                     {MTBFOld,MTBMAOld}
   OMHOld         := 6.34;
   IMHOld         := 9.01;
   FMHOld         := 13.13;
   MATLOld        := 1082.64;
   BCMOld         := 7.8;
   AVDLROld       := 1120.36;

   MTBFNew        := 80.00;
   MTBMANew       := 20.60;
   OMHNew         := 9.01;
   IMHNew         := 9.01;
   FMHNew         := 13.13;
   MATLNew        := 1082.64;
   BCMNew         := 4.0;
   AVDLRNew       := 1120.36;

   CF             := 5400000;

   LeadTime       := 60;
   INSTL          := 15;

   ST             := 6;
   Hor            := 180;
   Meth           := 1;
   Rep            := 100;
```

```
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{  This procedure uses the input data to compute the six unit
costs.                                                       }

Procedure SetUnitCost;
var OMHCostOld,IMHCostOld,FMHCostOld : real;
    OMHCostNew,IMHCostNew,FMHCostNew : real;

begin
   NOFNew  := NS*UPM/MTBFNew;
   NOMANew := NS*UPM/MTBMANew;

   OMHCostOld := OMHOld*MAML1Ratio*ML1Rate;
   IMHCostOld := IMHOld*MAML2Ratio*ML2Rate;
   FMHCostOld := FMHOld*ML2Rate;
   COMA := OMHCostOld+IMHCostOld;
   COF  := FMHCostOld+MATLOld;
   COAD := BCMOld*AVDLROld;

   OMHCostNew := OMHNew*MAML1Ratio*ML1Rate;
   IMHCostNew := IMHNew*MAML2Ratio*ML2Rate;
   FMHCostNew := FMHNew*ML2Rate;
   CNMA := OMHCostNew+IMHCostNew;
   CNF  := FMHCostNew+MATLNew;
   CNAD := BCMNew*AVDLRNew;
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure SetMission;

begin
   Mission[0]  := ' Run the model.';
   Mission[1]  := ' Data (Read file: 0, Simulate: 1)          :';
   Mission[2]  := ' Simu. para. (MuF, EtaF, Sig2F, CF )       :';
   Mission[3]  := ' Simu. para. (MuMA,EtaMA,Sig2MA,CMA)       :';
   Mission[4]  := ' Reading filename                          :';
   Mission[5]  := ' NO. of systems & Use/Mon (NS.UPM)         :';
   Mission[6]  := ' New MTBF & MTBMA (MTBFNew,MTBMANew)       :';
   Mission[7]  := ' O MH/MA    old & new (OMHOld,OMHNew)      :';
   Mission[8]  := ' I MH/MA    old & new (IMHOLd,IMHNew)      :';
   Mission[9]  := ' MH/F       old & new (FMHOld,FMHNew)      :';
   Mission[10] := ' MATL Cost   O & N (MATLOld,MATLNew)       :';
   Mission[11] := ' NO. of BCMs O & N (BCMOld,BCMNew)         :';
   Mission[12] := ' AVDLR Cost  O & N (AVDLROld,AVDLRNew)     :';
   Mission[13] := ' Fix cost (CF)                             :';
   Mission[14] := ' Lead Time & Kits/Mo (LeadTime,INSTL)      :';
   Mission[15] := ' Time start & horizon (ST,Hor)             :';
   Mission[16] := ' Method (Likelihood: 0, Bayesian: 1)       :';
```

```
   Mission[17] := ' Replication (Rep)                       :';
   Mission[18] := ' Reset to default values.';
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   This procedure allows the user to choose the desired
values for each parameter.
{   If Bayesian method (Meth = 1) is chosen, then starting
time will set to 6 automatically.                          }

Procedure SetParameter;
var Choise, D, M : integer;

{++++++++++++++++++++++++++++++++++++}

Procedure PrintMenu;
var Choise : integer;

begin
   writeln('If  you  want  to  change  the  values  of  any
parameter,');
   writeln('enter the number from nemu,');
   writeln('or enter 0 for running the model.');
   for Choise := 0 to 18 do
      writeln(Choise:2,'   ',Mission[Choise]);
end;

{++++++++++++++++++++++++++++++++++++}

Procedure ChooseItem;

begin
   repeat
      readln(Choise);
      if (Choise < 0) or (Choise > 18) then
         GotoXY(2,23);
   until (Choise >= 0) and (Choise <= 18);

   if (Choise <> 0) then
      write('Enter',Mission[Choise])
   else
      if (Data = 1) then
      begin
         writeln('How many data do you want
                 in the arrays (ET)? ');
         write('(Beware ET <= Mor-LeadTime) ');
         readln(ET);
      end;
end;

{++++++++++++++++++++++++++++++++++++}
```

```
Procedure SetInitial;

begin
   if Data = 0 then
   begin
      SetMuF         := 0;
      SetEtaF        := 0;
      SetSigmaSqrF   := 0;
      SetCF          := 0;
      SetMuMA        := 0;
      SetEtaMA       := 0;
      SetSigmaSqrMA  := 0;
      SetCMA         := 0;
   end
   else
   begin
      SetMuF         := SetMuF1;
      SetEtaF        := SetEtaF1;
      SetSigmaSqrF   := SetSigmaSqrF1;
      SetCF          := SetCF1;
      SetMuMA        := SetMuMA1;
      SetEtaMA       := SetEtaMA1;
      SetSigmaSqrMA  := SetSigmaSqrMA1;
      SetCMA         := SetCMA1;
   end;
   if Meth = 1 then
      ST := 6;
end;

{++++++++++++++++++++++++++++++++++++}

procedure PrintParameter;

begin
   GotoXY(45,5);  write(Data:10);
   GotoXY(45,6);  write(SetMuF:5:0,SetEtaF:5:0,
                        SetSigmaSqrF:5:0,SetCF:5);
   GotoXY(45,7);  write(SetMuMA:5:0,SetEtaMA:5:0,
                        SetSigmaSqrMA:5:0,SetCMA:5);
   GotoXY(45,8);  write(Infile:10);
   GotoXY(45,9);  write(NS:10:2,UPM:10:2);
   GotoXY(45,10); write(MTBFNew:10:2,MTBMANew:10:2);
   GotoXY(45,11); write(OMHOld:10:2,OMHNew:10:2);
   GotoXY(45,12); write(IMHOld:10:2,IMHNew:10:2);
   GotoXY(45,13); write(FMHOld:10:2,FMHNew:10:2);
   GotoXY(45,14); write(MATLOld:10:2,MATLNew:10:2);
   GotoXY(45,15); write(BCMOld:10:2,BCMNew:10:2);
   GotoXY(45,16); write(AVDLROld:10:2,AVDLRNew:10:2);
   GotoXY(45,17); write(CF:10:0);
   GotoXY(45,18); write(LeadTime:10,INSTL:10);
   GotoXY(45,19); write(ST:10,Hor:10);
```

```
   GotoXY(45,20); write(Meth:10);
   GotoXY(45,21); write(Rep:10);
   GotoXY(1,23);  write('? ');
end;

{++++++++++++++++++++++++++++++++++++}

begin {Procedure SetParameter}
   repeat
      ClrScr;
      PrintMenu;
      SetInitial;
      PrintParameter;
      ChooseItem;
      case Choise of
         0 : GotoXY(1,24);
         1 : begin
                readln(D);
                if (D = 0) or (D = 1) then
                   Data := D;
             end;
         2 : readln(SetMuF,SetEtaF,SetSigmaSqrF,SetCF);
         3 : readln(SetMuMA,SetEtaMA,SetSigmaSqrMA,SetCMA);
         4 : readln(Infile);
         5 : readln(NS,UPM);
         6 : readln(MTBFNew,MTBMANew);
         7 : readln(OMHOld,OMHNew);
         8 : readln(IMHOld,IMHNew);
         9 : readln(FMHOld,FMHNew);
        10 : readln(MATLOld,MATLNew);
        11 : readln(BCMOld,BCMNew);
        12 : readln(AVDLROld,AVDLRNew);
        13 : readln(CF);
        14 : readln(LeadTime,INSTL);
        15 : readln(ST,Hor);
        16 : begin
                readln(M);
                if (M = 0) or (M = 1) then
                   Meth := M;
             end;
        17 : readln(Rep);
        18 : DefaultParameter;
      end;
   until (Choise = 0);
end; {Procdure SetParameter}

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure WriteParameter;

begin
```

```
   writeln(Output,Mission[1]:10,Data:10);
   writeln(Output,Mission[2]:10,SetMuF:5:0,SetEtaF:5:0,
                  SetSigmaSqrF:5:0,SetCF:5);
   writeln(Output,Mission[3]:10,SetMuMA:5:0,SetEtaMA:5:0,
                  SetSigmaSqrMA:5:0,SetCMA:5);
   writeln(Output,Mission[4]:10,Infile:10);
   writeln(Output,Mission[5]:10,NS:10:2,UPM:10:2);
   writeln(Output,Mission[6]:10,MTBFNew:10:2,MTBMANew:10:2);
   writeln(Output,Mission[7]:10,OMHOld:10:2,OMHNew:10:2);
   writeln(Output,Mission[8]:10,IMHOld:10:2,IMHNew:10:2);
   writeln(Output,Mission[9]:10,FMHOld:10:2,FMHNew:10:2);
   writeln(Output,Mission[10]:10,MATLOld:10:2,MATLNew:10:2);
   writeln(Output,Mission[11]:10,BCMOld:10:2,BCMNew:10:2);
   writeln(Output,Mission[12]:10,AVDLROld:10:2,AVDLRNew:10:2);
   writeln(Output,Mission[13]:10,CF:10:0);
   writeln(Output,Mission[14]:10,LeadTime:10,INSTL:10);
   writeln(Output,Mission[15]:10,ST:10,Hor:10);
   writeln(Output,Mission[16]:10,Meth:10);
   writeln(Output,Mission[17]:10,Rep:10);
   writeln(Output);
end;

{+++PART2 : ESTIMATE THE DEGRADATION+++++++++++++++++++++}
{   These procedures estimate the time of onset of subsystem
degradation and the magnitude and evolution of the degradation
over time. It includes:
    ReadData
    BuildVec
    Mu_Eta              (both method),
    SigmaSqr_Sum        (both method),
    FindHatParameter1 (maximum likelihood),
    InitialSigmaSqr     (Bayesian),
    FindHatParameter2 (Bayesian).                              }

Procedure ReadData (var Input : text;
                    var XVec : RealVec;
                    var YVec : RealVec);
var I : integer;
    F, MA, TH : real;

{   This procedure reads data from a file, and transforms mean
time between failure (maintenance action) to number of
failures (maintenance actions).                               }

Procedure FindLength (Var Input : text;
                      var I : integer);
var F, MA : real;

begin
   reset (Input);
   I := 0;
```

```
   while not eof (Input) do
   begin
      while not eoln (Input) do
      begin
         read(Input,F,MA);
         I := I+1;
      end;
      readln(Input);
   end;
   ET := I;
end;

{++++++++++++++++++++++++++++++++++++}

Procedure Compare (XVec, YVec : RealVec;
                   I : integer);

begin
   if XVec[I] >= YVec[I] then
   begin
      if XVec[I] > Max then
         Max := XVec[I];
      if YVec[I] < Min then
         Min := YVec[I];
   end
   else
      if YVec[I] > Max then
         Max := YVec[I];
      if XVec[I] < Min then
         Min := XVec[I];
end;

{++++++++++++++++++++++++++++++++++++}

begin {Procedure ReadData}
   assign (Input,Infile);
   FindLength(Input,I);
   reset (Input);
   TH := NS*UPM;
   while not eof (Input) do
   begin
      while not eoln (Input) do
      begin
         read(Input,F,MA);
         if F <> 0 then
            XVec[I] := TH/F
         else
            XVec[I] := 0;
         if MA <> 0 then
            YVec[I] := TH/MA
         else
```

```
            YVec[I] := 0;
         Compare(XVec,YVec,I);
         I := I-1;
      end;
      readln(Input);
   end;
   close(Input);
end;  {Procedure ReadData}

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   These procedures use the given parameters (SetMuF,
SetEtaF, SetSigmaSqrF, SetCF, SetMuMA, SetEtaMA,
SetSigmaSqrMA, SetCMA) to generate two simulating data set.
    All parameters are used to generate number of failures
(maintenance actions) instead of mean time between failure
(maintenance action).                                      }

Function GenNormal : real;
var Uni : real;

Function GenUniform : real;
var U : real;

begin {Function GenUniform}
   U := Random;
   if U > 0 then
      GenUniform := U
   else
      GenUniform := Random;
end;  {Function GenUniform}

begin {Function GenNormal}
   Uni := GenUniform;
   GenNormal := Sqrt((-Ln(Uni)*2))*Cos(2*Pi*Random);
end;  {Function GenNormal}

{++++++++++++++++++++++++++++++++++++}

Procedure BuiidVec (T, SetC : integer;
                    SetSigmaSqr, SetMu, SetEta : real;
                    var XVec : RealVec);
var I, DF, D : integer;
    NorF, Nor, SigmaF, Sigma : real;

begin {Procedure BuildVec}
   Sigma := sqrt(SetSigmaSqr);
   for I := 1 to T do
   begin
      Nor := GenNormal;
      if I <= SetC then
         D := 0
```

```
      else
         D := I-SetC;
      XVec[I] := SetMu+(D*SetEta)+(Sigma*Nor);
   end;
end;  {Procedure BuildVec}

{++++++++++++++++++++++++++++++++++}

Procedure BuildVec1 (T, SetC : integer;
                     SetSigmaSqr, SetMu, SetEta : real;
                     var XVec : RealVec);
var I, DF, D : integer;
    NorF, Nor, SigmaF, Sigma : real;

begin {Procedure BuildVec1}
   Sigma := sqrt(SetSigmaSqr);
   for I := 1 to T do
   begin
      Nor := GenNormal;
      if I <= SetC then
         D := 0
      else
         D := I-SetC;
      XVec[I] := SetMu+(D*SetEta)+(Sigma*Nor);
      if XVec[I] > Max then
         Max := XVec[I]
      else
         if XVec[I] < Min then
            Min := XVec[I];
   end;
end;  {Procedure BuildVec1}

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   This procedure calculates the mean value of array before
the trend, the occuring time of the trend, and the slope of
the trend.                                                   }

Procedure Mu_Eta (T, C : integer;
                  var Phi1 : real;
                  var Phi2 : real;
                  XVec : RealVec;
                  var Mu : RealVec;
                  var Eta : RealVec);
var X1, X2, Phi : real;

{++++++++++++++++++++++++++++++++++++}

Function BarX1 (T : integer;
                XVec : RealVec) : real;
var I   : integer;
    Sum : real;
```

```
begin {Function BarX1}
   Sum := 0;
   for I := 1 to T do
      Sum := Sum+XVec[I];
   BarX1 := Sum/T;
end;  {Function BarX1}

{+++++++++++++++++++++++++++++++++++++}

Function BarX2 (T, C : integer;
                XVec : RealVec) : real;
Var I   : integer;
    Sum : real;

begin {Function BarX2}
   Sum := 0;
   for I := (C+1) to T do
      Sum := Sum+(XVec[I]*(I-C));
   BarX2 := Sum/T;
end;  {Function BarX2}

{+++++++++++++++++++++++++++++++++++++}

Function BarPhi1 (T, C : integer) : real;
var I   : integer;
    Sum : real;

begin {Function BarPhi1}
   Sum := 0;
   for I := 1 to (T-C) do
      Sum := Sum+I;
   BarPhi1 := Sum/T;
end;  {Function BarPhi1}

{+++++++++++++++++++++++++++++++++++++}

Function BarPhi2 (T, C : integer) : real;
var I   : integer;
    Sum, Temp : real;

begin {Function BarPhi2}
   Sum := 0;
   for I := 1 to (T-C) do
      Sum := Sum+(I*I);
   BarPhi2 := Sum/T;
end;  {Function BarPhi2}

{+++++++++++++++++++++++++++++++++++++}

begin {Procedure Mu_Eta}
   X1 := BarX1(T,XVec);
```

```
   if C < T then
   begin
      X2      := BarX2(T,C,XVec);
      Phi1    := BarPhi1(T,C);
      Phi2    := BarPhi2(T,C);
      Phi     := Phi2-(Phi1*Phi1);
      Mu[C]   := ((Phi2*X1)-(Phi1*X2))/Phi;
      Eta[C]  := (X2-(Phi1*Mu[C]))/Phi2;
   end
   else
   begin
      Mu[C] := X1;
      Eta[C] := 0;
   end;
end;  {Procedure Mu_Eta}

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Function SigmaSqr_Sum (T, C : integer;
                       XVec, Mu, Eta : RealVec) : real;
var I, D : integer;
    Temp, Sum : real;

begin {Function SigmaSqr_Sum}
   Sum := 0;
   for I := 1 to T do
   begin
      if I <= C then
         D := 0
      else
         D := (I-C);
      Temp := XVec[I]-Mu[C]-(D*Eta[C]);
      Sum := Sum+(Temp*Temp);
   end;
   SigmaSqr_Sum := Sum;       {Likelihood : /T, Bayesian :
/T-1}
end;  {Function SigmaSqr_Sum}

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure FindHatParameter1(T : integer;
                            var HatC : integer;
                            XVec : RealVec;
                            var Mu : RealVec;
                            var Eta : RealVec;
                            var SigmaSqr : RealVec;
                            var HatMu : real;
                            var HatEta : real);
var C  : integer;
    Phi1, Phi2, S, MinS, HSS : real;
```

```
begin {Procedure FindHatParameter1}
   MinS := 0;
   for C := 1 to T do
   begin
      Mu_Eta(T,C,Phi1,Phi2,XVec,Mu,Eta);
      SigmaSqr[C] := SigmaSqr_Sum(T,C,XVec,Mu,Eta)/T;
      if SigmaSqr[C] > 0 then
         S := 1+ln(SigmaSqr[C])
      else
         S := -1E+10;
      if (MinS = 0) or (MinS >= S) then
      begin
         MinS := S;
         HatC := C;
      end;
   end;
   HatMu := Mu[HatC];
   HatEta := Eta[HatC];
end;  {Procedure FindHatParameter1}

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure InitialSigmaSqr (XVec : RealVec;
                           var Mu : RealVec;
                           var Eta : RealVec;
                           var SigmaSqr : RealVec);
var I : integer;
    Sum : real;

begin {Procedure InitialSigmaSqr}
    Sum := 0;
    for I := 1 to 5 do
       Sum := Sum+XVec[I];
    Mu[5] := Sum/5;
    Eta[5] := 0;
    SigmaSqr[5] := SigmaSqr_Sum(5,5,XVec,Mu,Eta)/4;
end;  {Procedure InitialSigmaSqr}

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure FindHatParameter2 (T : integer;
                             XVec : RealVec;
                             var Mu : RealVec;
                             var Eta : RealVec;
                             var SigmaSqr : RealVec;
                             var VSqr : RealVec;
                             var RSqr : RealVec;
                             var RhoSqr : RealVec;
                             var PiStar : RealVec);
var C   : integer;
    Sum, Sum1, Sum2 : real;
```

```
    Phi1, Phi2, Value : real;
    Coef, Sig2, KStar : RealVec;

{++++++++++++++++++++++++++++++++++++}

Procedure V_R_RhoSqr (T, C : integer;
                      Phi1, Phi2  : real;
                      SigmaSqr : RealVec;
                      var VSqr : RealVec;
                      var RSqr : RealVec;
                      var RhoSqr : RealVec;
                      var Coef : RealVec);
var Phi : real;

begin {Procedure V_R_RhoSqr}
   if C < T then
   begin
      Phi := Phi2-(Phi1*Phi1);
      VSqr[C]   := SigmaSqr[T-1]/T/Phi;
      RSqr[C]   := VSqr[C]*Phi2;
      RhoSqr[C] := Phi1*Phi1/Phi2;
      Coef[C]   := 2*Pi*Sqrt((1-RhoSqr[C])*RSqr[C]*VSqr[C]);
   end
   else
   begin
      RSqr[C]   := SigmaSqr[T-1]/T;
      Coef[C]   := Sqrt(2*Pi*RSqr[C]);
      VSqr[C]   := 0;
      RhoSqr[C] := 0;
   end;
end;  {Procedure V_R_RhoSqr}

{++++++++++++++++++++++++++++++++++++}

Function Beta (K, T : real): real;
var N, D : real;

{   This function computes the prior function combined with
geometric probability function.
    It is given that both coefficients of beta (prior)
function are 1.                                             }

begin {Function Beta}
   if K = 1 then
      Beta := 0.5
   else
      if K =2 then
         Beta := 1/6
      else
         if K <= T then
            Beta := 1/(K*(K+1))
```

```
          else
             Beta := 1/K;
end;  {Function Beta}

{++++++++++++++++++++++++++++++++++++}

begin {Procedure FindHatParameter2}
   Sum := 0;
   for C := 1 to (T+1) do
   begin
      Mu_Eta(T,C,Phi1,Phi2,XVec,Mu,Eta);

V_R_RhoSqr(T,C,Phi1,Phi2,SigmaSqr,Vsqr,RSqr,RhoSqr,Coef);
      Sig2[C] := SigmaSqr_Sum(T,C,XVec,Mu,Eta);
      KStar[C] := Sig2[C]/(2*SigmaSqr[T-1]);
      Value := 50-KStar[C];
      if Value >= -86 then
         PiStar[C] := exp(Value)*Beta(C,T)*Coef[C]
      else
         PiStar[C] := 0;
{        PiStar[C] := Beta(C,T)*exp(-KStar[C])*Coef[C];}
      Sum := Sum+PiStar[C];
   end;

   Sum1 := 0;
   for C := 1 to (T+1) do
   begin
      PiStar[C] := PiStar[C]/Sum;
      if PiStar[C] < 1E-6 then
         Pistar[C] := 0;
      Sum1 := Sum1+PiStar[C];
   end;

   Sum2 := 0;
   for C := 1 to (T+1) do
   begin
      PiStar[C] := PiStar[C]/Sum1;
      Sum2 := Sum2+(PiStar[C]*Sig2[C]/(T-1));
   end;
   SigmaSqr[T] := Sum2;
end;  {Procedure FindHatParameter2}

{+++PART3 : COMPUTE THE COSTS & SEARCH BEST CHANGE TIME ++++}
{   These procedures compare the cost of remaining with the
current subsystem with the cost of investing in the upgraded
subsystem to obtain a best time to invest in the upgraded
subsystem. It includes:
    HatCost_New      (both method),
    HatCost_NewAD    (both method),
    HatCost_Old      (both method),
    HatCost_OldAD    (both method),
```

```
    FindCostNew      (Bayesian),
    FindCostOld      (Bayesian),
    Mean_Variance    (Bayesian),
    SearchTime       (both method),
    FindChangeTime1  (maximum likelihood),
    FindChangeTime2  (Bayesian),
    Uncertainty1     (maximum likelihood),
    Uncertainty2     (Bayesian).                                }

Procedure Install (H, T, Tau : integer;
                   var N : integer;
                   var PartialSum : real);

begin {Procedure Install}
   N := trunc(NS/INSTL);
   if N > (H-(T+Tau)) then
      N := H-(T+Tau);
   PartialSum := (N*(N+1)/2)*(INSTL/NS);
end;  {Procedure Install}

{+++++++++++++++++++++++++++++++++++++}

Function HatCost_NewAD (H, T, Tau, N : integer;
                        PartialSum, CO, CN : real): real;
var NewPart, OldPart : real;

begin {Function HatCost_NewAD}
   NewPart := CN*(PartialSum-N+H-(T+Tau));
   OldPart := CO*(Tau+1+N-PartialSum);
   HatCost_NewAD := OldPart+NewPart;
end;  {Function HatCost_NewAD}

{+++++++++++++++++++++++++++++++++++++}

Function HatCost_New (H, T, Tau, HatC, N : integer;
                      PartialSum, HatMu, HatEta : real;
                      CO, CN, NONew : real): real;

var S, D : integer;
    Sum, Partial, NewPart, OldPart : real;

begin {Function HatCost_New}
   NewPart := (CN*NONew)*(PartialSum-N+H-(T+Tau));
   if HatC >= T then
      OldPart := (CO*HatMu)*(Tau+1+N-PartialSum)
   else
   begin
      D := T-HatC;
      Sum := 0;
      for S := 0 to (Tau+N) do
      begin
```

```
         if S <= Tau then
            Partial := 1
         else
            Partial := 1-(S-Tau)*(INSTL/NS);
         Sum := Sum+(HatMu+HatEta*(S+D))*Partial;
      end;
      OldPart := CO*Sum;
   end;
   HatCost_New := OldPart+NewPart;
end;  {Function HatCost_New}

{++++++++++++++++++++++++++++++++++++++}

Function HatCost_OldAD (H, T : integer;
                        CO : real) : real;

begin {Function HatCost_OldAD}
   HatCost_OldAD := CO*(H-(T-1))
end;  {Function HatCost_OldAD}

{++++++++++++++++++++++++++++++++++++++}

Function HatCost_Old (H, T, HatC: integer;
                      HatMu, HatEta, CO : real) : real;
var S, D : integer;
    Sum : real;

begin {Function HatCost_Old}
   if HatC > T then
      HatCost_Old := CO*HatMu*(H-(T-1))
   else
   begin
      D := T-HatC;
      Sum := 0;
      for S := 0 to (H-T) do
         Sum := Sum+HatMu+HatEta*(S+D);
   end;
   HatCost_Old := CO*Sum;
end;  {Function HatCost_Old}

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Function FindCostNew (H, T, Tau, N : integer;
                      PartialSum, CO, CN, NONew : real;
                      Mu, Eta : RealVec;
                      PiStar : RealVec) : real;
var Sum, HCN, HatMu, HatEta : real;
    C : integer;

begin {Function FindCostNew}
   Sum := 0;
```

```
   for C := 1 to (T-LeadTime+1) do
   begin
      HatMu := Mu[C];
      HatEta := Eta[C];
      if PiStar[C] <> 0 then
         HCN := HatCost_New(H,T,Tau,C,N,PartialSum,
                        HatMu,HatEta,CO,CN,NONew)*PiStar[C]
      else
         HCN := 0;
      Sum := Sum+HCN;
   end;
   FindCostNew := Sum;
end;  {Function FindCostNew}

{++++++++++++++++++++++++++++++++++++}

Function FindCostOld (H, T : integer; CO : real;
                      Mu, Eta : RealVec;
                      PiStar : RealVec) : real;
var Sum, HCO, HatMu, HatEta : real;
    C : integer;

begin {Function FindCostOld}
   Sum := 0;
   for C := 1 to (T-LeadTime+1) do
   begin
      HatMu := Mu[C];
      HatEta := Eta[C];
      if PiStar[C] <> 0 then
         HCO := HatCost_Old(H,T,C,HatMu,HatEta,CO)*PiStar[C]
      else
         HCO := 0;
      Sum := Sum+HCO;
   end;
   FindCostOld := Sum;
end;  {Function FindCostOld}

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure Mean_Variance (H, T, Tau, N : integer;
                         PartialSum, CO, CN, NONew : real;
                         Mu, Eta : RealVec;
                         RSqr, VSqr, RhoSqr : RealVec;
                         PiStar : RealVec;
                         var CostMean : real;
                         var CostVariance : real);
var C : integer;
    Sum, CostVar1, CostVar2 : real;
    MuCo, ConstCo : real;
    EtaCo, CondMean : RealVec;
```

```
{++++++++++++++++++++++++++++++++++++}

Function ConstCoef (H, T, Tau, N : integer;
                    PartialSum, CN, NONew : real): real;

begin {Function ConstCoef}
   ConstCoef := (CN*NONew)*(PartialSum-N+H-(T+Tau));
end;  {Function ConstCoef}

{++++++++++++++++++++++++++++++++++++}

Function MuCoef (H, T, Tau, N : integer;
                 CO : real): real;
var S : integer;
    Sum, Partial : real;

begin {Function MuCoef}
   Sum := 0;
   for S := 0 to (Tau+N) do
   begin
      if S <= Tau then
         Partial := 1
      else
         Partial := 1-(S-Tau)*(INSTL/NS);
      Sum := Sum-Partial;
   end;
   for S := 0 to (H-T) do
      Sum := Sum+1;
   MuCoef := CO*Sum;
end;  {Function MuCoef}

{++++++++++++++++++++++++++++++++++++}

Function EtaCoef (H, T, Tau, HatC, N : integer;
                  CO : real): real;

var S, D : integer;
    Sum, Partial : real;

begin {Function EtaCoef}
   if HatC >= T then
      EtaCoef := 0
   else
   begin
      D := T-HatC;
      Sum := 0;
      for S := 0 to (Tau+N) do
      begin
         if S <= Tau then
            Partial := 1
         else
```

```
            Partial := 1-(S-Tau)*(INSTL/NS);
         Sum := Sum-(S+D)*Partial;
      end;
      for S := 0 to (H-T) do
         Sum := Sum+(S+D);
   end;
   EtaCoef := CO*Sum;
end;  {Function EtaCoef}

{+++++++++++++++++++++++++++++++++++}

begin {Function Mean_Variance}
   MuCo := MuCoef(H,T,Tau,N,CO);
   ConstCo := ConstCoef(H,T,Tau,N,PartialSum,CN,NONew);
   Sum := 0;
   for C := 1 to (T-LeadTime+1) do
   if PiStar[C] <> 0 then
   begin
      EtaCo[C] := EtaCoef(H,T,Tau,C,N,CO);
      CondMean[C] := MuCo*Mu[C]+EtaCo[C]*Eta[C]-ConstCo;
      Sum := Sum+CondMean[C]*PiStar[C];
   end;
   CostMean := Sum;

   Sum := 0;
   for C := 1 to (T-LeadTime+1) do
   if PiStar[C] <> 0 then
      Sum := Sum+sqr(CondMean[C]-CostMean)*PiStar[C];
   CostVar2 := Sum;

   Sum := 0;
   for C := 1 to (T-LeadTime+1) do
   if PiStar[C] <> 0 then
      Sum := Sum+(sqr(MuCo)*RSqr[C]+sqr(EtaCo[C])*VSqr[C]
                 -2*sqrt(RhoSqr[C]*RSqr[C]*VSqr[C])
                 *MuCo*EtaCo[C])*PiStar[C];
   CostVar1 := Sum;

   CostVariance := CostVar1+CostVar2;
end;  {Function Mean_Variance}

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure SearchTime (H, N : integer;
                      HC : real;
                      var Tau : integer;
                      var CT : integer;
                      var MinC : real);

begin {Procedure SearchTime}
   if HC > MinC then
```

```
   begin
      if MinC > 0 then
         CT := H
      else
         CT := CT-1;
      Tau := H;
   end
   else
   begin
      MinC := HC;
      if CT = (H-LeadTime-N) then
      begin
         if MinC > 0 then
            CT := H;
         Tau := H;
      end;
   end;
end;  {Procedure SearchTime}

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   Procedures "FindChangeTime1" and "FindChangeTime2"
estimate the best time for subsystem upgrade.              }

Procedure FindChangeTime1 (H, T, HatCF, HatCMA : integer;
                           HatMuF, HatEtaF : real;
                           HatMuMA, HatEtaMA : real);
var N, Tau, CT, T1 : integer;
    HC, MinC, PartialSum, CostNew, CostOld, CostFix : real;

begin {Procedure FindChangeTime1}
   Tau := 0;
   T1 := T-LeadTime;
   Install(H,T,Tau,N,PartialSum);
   CostFix := CF;
   CostNew := HatCost_New(H,T,Tau,HatCF,N,PartialSum,
                    HatMuF,HatEtaF,COF,CNF,NOFNew)
             +HatCost_New(H,T,Tau,HatCMA,N,PartialSum,
                    HatMuMA,HatEtaMA,COMA,CNMA,NOMANew)
             +HatCost_NewAD(H,T,Tau,N,PartialSum,COAD,CNAD)
             +CostFix;
   CostOld := HatCost_Old(H,T,HatCF,HatMuF,HatEtaF,COF)
             +HatCost_Old(H,T,HatCMA,HatMuMA,HatEtaMA,COMA)
             +HatCost_OldAD(H,T,COAD);
   MinC := CostNew-CostOld;

   if MinC > CF then
      CTVec[T1] := H
   else
   begin
      while Tau <= (H-T) do
      begin
```

```
         Tau := Tau+1;
         CT := T1+Tau;
         if CT <= (H-LeadTime-N) then
         begin
            if H = (T+Tau) then
               CostFix := 0;
            CostNew := HatCost_New(H,T,Tau,HatCF,N,PartialSum,
                                   HatMuF,HatEtaF,COF,CNF,NOFNew)
                       +HatCost_New(H,T,Tau,HatCMA,N,
                                    PartialSum,HatMuMA,HatEtaMA,
                                    COMA,CNMA,NOMANew)
                       +HatCost_NewAD(H,T,Tau,N,PartialSum,
                                    COAD,CNAD)
                       +CostFix;
            HC := CostNew-CostOld;
            SearchTime(H,N,HC,Tau,CT,MinC);
         end
         else
            CT := H;
      end;
      CTVec[T1] := CT;
   end;
   write(T1:5,CTVec[T1]:8);
   write(Output,T1:5,CTVec[T1]:8);
end;  {Procedure FindChangeTime1}

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure FindChangeTime2 (H, T : integer;
                           MuF, EtaF : RealVec;
                           MuMA, EtaMA : RealVec;
                           PiStarF, PiStarMA : RealVec);
var N, Tau, CT, T1 : integer;
    HC, MinC, PartialSum, CostNew, CostOld, CostFix : real;

begin {Procedure FindChangeTime2}
   Tau := 0;
   T1 := T-LeadTime;
   Install(H,T,Tau,N,PartialSum);
   CostFix := CF;
   CostNew := FindCostNew(H,T,Tau,N,PartialSum,
                          COF,CNF,NOFNew,MuF,EtaF,PiStarF)
              +FindCostNew(H,T,Tau,N,PartialSum,
                          COMA,CNMA,NOMANew,MuMA,EtaMA,PiStarMA)
              +HatCost_NewAD(H,T,Tau,N,PartialSum,COAD,CNAD)
              +CostFix;
   CostOld := FindCostOld(H,T,COF,MuF,EtaF,PiStarF)
              +FindCostOld(H,T,COMA,MuMA,EtaMA,PiStarMA)
              +HatCost_OldAD(H,T,COAD);
   MinC := CostNew-CostOld;
```

```
   if MinC > CF then
      CTVec[T1] := H
   else
   begin
      while Tau <= (H-T) do
      begin
         Tau := Tau+1;
         CT := T1+Tau;
         if CT <= (H-LeadTime-N) then
         begin
            if H = (T+Tau) then
               CostFix := 0;
            CostNew := FindCostNew(H,T,Tau,N,PartialSum,COF,
                                   CNF,NOFNew,MuF,EtaF,PiStarF)
                       +FindCostNew(H,T,Tau,N,PartialSum,COMA,
                                   CNMA,NOMANew,MuMA,EtaMA,
                                   PiStarMA)
                       +HatCost_NewAD(H,T,Tau,N,PartialSum,
                                   COAD,CNAD)
                       +CostFix;
            HC := CostNew-CostOld;
            SearchTime(H,N,HC,Tau,CT,MinC);
         end
         else
            CT := H;
      end;
      CTVec[T1] := CT;
   end;
   write(T1:5,CTVec[T1]:8);
   write(Output,T1:5,CTVec[T1]:8);
end;  {Procedure FindChangeTime2}

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   Procedures "Uncertainty1" and "Uncertainty2" compute the
MEAN, SD, and 2 SD bounds of cost advantage of upgrade for
the estimated best upgrade time.                              }

Procedure Uncertainty1 (H, T, R, HatCF, HatCMA : integer;
                        HatMuF, HatEtaF : real;
                        HatMuMA, HatEtaMA : real);
var PartialSum, CostNew, CostOld, Sum, SD : real;
    N, Tau : integer;

begin {Procedure Uncertainty1}
   Tau := CTVec[Time]-Time;
   if CTVec[Time] = Hor then
      Tau := 0;
   CostOld := HatCost_Old(H,T,HatCF,HatMuF,HatEtaF,COF)
             +HatCost_Old(H,T,HatCMA,HatMuMA,HatEtaMA,COMA)
             +HatCost_OldAD(H,T,COAD);
   Install(H,T,Tau,N,PartialSum);
```

```
   CostNew := HatCost_New(H,T,Tau,HatCF,N,PartialSum,
                          HatMuF,HatEtaF,COF,CNF,NOFNew)
              +HatCost_New(H,T,Tau,HatCMA,N,PartialSum,
                          HatMuMA,HatEtaMA,COMA,CNMA,NOMANew)
              +HatCost_NewAD(H,T,Tau,N,PartialSum,COAD,CNAD)
              +CF;
   Gain[R] := CostOld-CostNew;
   Total := Total+Gain[R];

   if R = Rep then
   begin
      Sum := 0;
      Mean[Time] := Total/Rep;
      for R := 1 to Rep do
         Sum := Sum+sqr(Gain[R]-Mean[Time]);
      SD := sqrt(Sum/(Rep-1));
      LB[Time] := Mean[Time]-2*SD;
      UB[Time] := Mean[Time]+2*SD;
      if Min > LB[Time] then
         Min := LB[Time];
      if Max < UB[Time] then
         Max := UB[Time];
      writeln('     ',Mean[Time]:10,'   ',
              SD:10,'   ',LB[Time]:10,'   ',UB[Time]:10);
      writeln(Output,'     ',Mean[Time]:10,'   ',
              SD:10,'   ',LB[Time]:10,'   ',UB[Time]:10);
   end;
end;  {Procedure Uncertainty1}

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure Uncertainty2 (H, T : integer;
                        MuF, EtaF, MuMA, EtaMA : RealVec;
                        RSqrF, VSqrF, RhoSqrF : RealVec;
                        RSqrMA, VSqrMA, RhoSqrMA : RealVec;
                        PiStarF, PiStarMA : RealVec);
var PartialSum, MeanF, MeanMA : real;
    VarianceF, VarianceMA, SD : real;
    N, Tau : integer;

begin {Procedure Uncertainty2}
   Tau := CTVec[Time]-Time;
   if CTVec[Time] = Hor then
      Tau := 0;
   Install(H,T,Tau,N,PartialSum);
   Mean_Variance(H,T,Tau,N,PartialSum,COF,CNF,NOFNew,
                 MuF,EtaF,RSqrF,VSqrF,RhoSqrF,PiStarF,
                 MeanF,VarianceF);
   Mean_Variance(H,T,Tau,N,PartialSum,COMA,CNMA,NOMANew,
                 MuMA,EtaMA,RSqrMA,VSqrMA,RhoSqrMA,PiStarMA,
                 MeanMA,VarianceMA);
```

```
   SD := Sqrt(VarianceF+VarianceMA);
   Mean[Time] := MeanF+MeanMA-CF
                 -HatCost_NewAD(H,T,Tau,N,PartialSum,COAD,CNAD)
                 +HatCost_OldAD(H,T,COAD);
   LB[Time] := Mean[Time]-2*SD;
   UB[Time] := Mean[Time]+2*SD;
   if Min > LB[Time] then
      Min := LB[Time];
   if Max < UB[Time] then
      Max := UB[Time];
   writeln('     ',Mean[Time]:10,'  ',
           SD:10,'  ',LB[Time]:10,'  ',UB[Time]:10);
   writeln(Output,'     ',Mean[Time]:10,'  ',
           SD:10,'  ',LB[Time]:10,'  ',UB[Time]:10);
end;  {Procedure Uncertainty2}

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   These procedures reset the values for assessing
uncertainty.
    1 for maximum likelihood,
    2 for Bayesian.                                        }

Procedure ResetParameter1;

begin {Procedure ResetParameter1}
   SetCF := HatCF;
   SetMuF := HatMuF;
   SetEtaF := HatEtaF;
   SetSigmaSqrF := SigmaSqrF[SetCF];

   SetCMA := HatCMA;
   SetMuMA := HatMuMA;
   SetEtaMA := HatEtaMA;
   SetSigmaSqrMA := SigmaSqrMA[SetCMA];
end;  {Procedure ResetParameter1}

{++++++++++++++++++++++++++++++++++++}

Procedure ResetParameter2;
var T : integer;

begin {Procedure ResetParameter2}
   PiDistF[1] := PiStarF[1];
   for T := 2 to (Time+1) do
      PiDistF[T] := PiDistF[T-1]+PiStarF[T];

   PiDistMA[1] := PiStarMA[1];
   for T := 2 to (Time+1) do
      PiDistMA[T] := PiDistMA[T-1]+PiStarMA[T];
end;  {Procedure ResetParameter2}
```

```
{+++PART4 : GENERATE DATA GRAPH +++++++++++++++++++++++++++}
{   These procedures plot the graph of the mean number of
failures and mean number of maintenance actions.          }

Procedure GenDataGraph;
var GraphDriver, GraphMode : integer;
    YScale, XScale : real;

{+++++++++++++++++++++++++++++++++++}

Procedure DrawAxis;
var S : String;

begin {Procedure DrawAxis}
   SetColor(White);
   OutTextXY(10,10,' NO. of Failures');
   OutTextXY(10,20,'(Maintenance Actions)');
   OutTextXY(250,10,'Mean number of Failures');
   OutTextXY(250,20,'Mean number of Maintenance Actions');
   Line(30,30,30,430);
   Line(30,430,630,430);
end;  {Procedure DrawAxis}

{+++++++++++++++++++++++++++++++++++}

Procedure SetYScale (YScale : real);
var I, Y : integer;
    M, Scale, Length, Units : real;
    S : string;

begin {Procedure SetYScale}
   M := 10;
   while (Max/M) > 10 do
      M := M*10;
   Scale := M/2;
   Length := 0;
   Units := 0;
   while Length > -Min do
   begin
      Length := Length-Scale;
      Units := Units+0.5;
      if Units > 10 then
         units := Units-10;
   end;
   I := round(frac(Units)) ;
   while Length > -Max do
   begin
      Y := round((Length+Min)*YScale);
      Line(30,430+Y,630,430+Y);
      if (I mod 2) = 0 then
      begin
```

```
            Str(Units:1:0,S);
            OutTextXY(15,430+Y,S);
         end;
         Length := Length-Scale;
         Units := Units+0.5;
         I := I+1;
      end;
      Str(round(M),S);
      OutTextXY(30,470,'Y-Axis scale');
      OutTextXY(130,470,S);
   end;  {Procedure SetYScale}

   {++++++++++++++++++++++++++++++++++}

   Procedure SetXScale (XScale : real);
   var T, T1, X : integer;
       S : string;

   begin {Procedure SetXScale}
      T  := 5;
      while T <= ET do
      begin
         X := round((T-1)*XScale);
         Line(30+X,30,30+X,430);
         if (T mod 10) = 0 then
         begin
            Str(T,S);
            OutTextXY(15+X,440,S);
         end;
         T := T+5;
      end;
      OutTextXY(300,460,'Month');
   end;  {Procedure SetXScale}

   {++++++++++++++++++++++++++++++++++}

   Procedure DrawLine (YScale, XScale, Mu, Eta : real;
                       C : integer);
   var X1, Y1, Y2 : integer;

   begin {Procedure DrawCurve}
      X1 := 30+round((C-1)*XScale);
      Y1 := 430-round((Mu-Min)*YScale);
      Line(30,Y1,X1,Y1);
      if ET > C then
      begin
         Y2 := 430-round((Mu+Eta*(ET-C)-Min)*YScale);
         Line(X1,Y1,630,Y2);
      end;
   end;  {Procedure DrawCurve}
```

```
{+++++++++++++++++++++++++++++++++++}

Procedure DrawCurve (YScale, XScale : real; XVec : RealVec);
var X1, X2, T : integer;
    YM1, YM2 : integer;

begin {Procedure DrawCurve}
   T := 1;
   X1 := 30;
   YM1 := 430-round((XVec[T]-Min)*YScale);
   for T := 2 to ET do
   begin
      X2 := round((T-1)*Xscale+30);
      YM2 := 430-round((XVec[T]-Min)*YScale);
      Line(X1,YM1,X2,YM2);
      OutTextXY(X2-3,YM2-5,'.');
      X1 := X2;
      YM1 := YM2;
   end;
end;  {Procedure DrawCurve}

{+++++++++++++++++++++++++++++++++++}

begin {Procedure GenDataGraph} {Verify Driver, Mode, and path}
   GraphDriver := VGA;        {adjust to the acceptable values}
   GraphMode   := VGAHi;
   InitGraph(GraphDriver,GraphMode,'a:\bgi\');
   SetBKColor(Black);
   ClearDevice;
   DrawAxis;
   YScale := 400/(Max-Min);
   XScale := 600/(ET-1);
   SetLineStyle(DottedLn,0,NormWidth);
   SetYScale(YScale);
   SetXScale(XScale);

   if Data = 1 then
   begin
      SetLineStyle(DashedLn,0,NormWidth);
      SetColor(Red);
      DrawLine(YScale,XScale,SetMuF,SetEtaF,SetCF);
      SetColor(Green);
      DrawLine(YScale,XScale,SetMuMA,SetEtaMA,SetCMA);
   end;
   SetLineStyle(SolidLn,0,NormWidth);
   SetColor(Red);
   DrawCurve(YScale,XScale,NOFVec);
   Line(550,15,580,15);
   SetColor(Green);
   DrawCurve(YScale,XScale,NOMAVec);
   Line(550,25,580,25);
```

```
   SetColor(White);
   OutTextXY(400,470,'*** Press <Enter> to exit ***');
   readln;
   CloseGraph;
end;  {Procedure GenDataGraph}

{+++PART5 : GENERATE COST GRAPH ++++++++++++++++++++++++++++}
{  These procedures plot the graph of the mean values and two
standard deviation bounds for cost advantage of upgrade.    }

Procedure GenGraph;
var GraphDriver, GraphMode : integer;
    YScale, XScale : real;
    YZero : integer;

{++++++++++++++++++++++++++++++++++++++}

Procedure DrawAxis;
var S : String;

begin {Procedure DrawAxis}
   SetColor(White);
   OutTextXY(10,10,'Cost advantage');
   OutTextXY(10,20,'  of upgrade  ');
   OutTextXY(250,10,'Assessing uncertainty');
   OutTextXY(250,20,'from time      to');
   Str(ST,S);
   OutTextXY(350,20,S);
   Str(ET,S);
   OutTextXY(400,20,S);
   Line(30,30,30,430);
   Line(30,430,630,430);
end;  {Procedure DrawAxis}

{++++++++++++++++++++++++++++++++++++++}

Procedure SetYScale (YScale : real; YZero : integer);
var I, Y ,E: integer;
    CostDiff, M, Scale, Length, Units : real;
    S : string;

begin {Procedure SetYScale}
   CostDiff := Max-Min;
   M := 10;
   E := 1;
   while (CostDiff/M) > 10 do
   begin
      M := M*10;
      E := E+1;
   end;
   Scale := M/2;
```

```
   Length := 0;
   Units := 0;
   I := 0;
   while Length > -Max do
   begin
      Y := round(Length*YScale);
      Line(30,YZero+Y,630,YZero+Y);
      if (I mod 2) = 0 then
      begin
         Str(Units:1:0,S);
         OutTextXY(15,YZero+Y,S);
      end;
      Length := Length-Scale;
      Units := Units+0.5;
      I := I+1;
   end;

   Scale := M/2;
   Length := Scale;
   Units := -0.5;
   I := 1;
   while Length < -Min do
   begin
      Y := round(Length*YScale);
      Line(30,YZero+Y,630,YZero+Y);
      if (I mod 2) = 0 then
      begin
         Str(Units:1:0,S);
         OutTextXY(15,YZero+Y,S);
      end;
      Length := Length+Scale;
      Units := Units-0.5;
      I := I+1;
   end;
   Str(E,S);
   OutTextXY(30,470,'Y-Axis scale E+');
   OutTextXY(150,470,S);
end;  {Procedure SetYScale}

{++++++++++++++++++++++++++++++++++++}

Procedure SetXScale (XScale : real);
var T, T1, X : integer;
    S : string;

begin {Procedure SetXScale}
   T  := ST;
   T1 := ST mod 5;
   if T1 <> 0 then
      T := T+5-T1;
   while T <= ET do
```

```
   begin
      X := round((T-ST)*XScale);
      Line(30+X,30,30+X,430);
      if (T mod 10) = 0 then
      begin
         Str(T,S);
         OutTextXY(15+X,440,S);
      end;
      T := T+5;
   end;
   OutTextXY(300,460,'Month');
end;  {Procedure SetXScale}

{++++++++++++++++++++++++++++++++++++++}

Procedure DrawCurve (YScale, XScale : real);
var X1, X2, T : integer;
    YM1, YM2, YUB1, YUB2, YLB1, YLB2 : integer;

begin {Procedure DrawCurve}
   T := ST;
   X1 := 30;
   YM1 := YZero-round(Mean[T]*YScale);
   YUB1 := YZero-round(UB[T]*YScale);
   YLB1 := YZero-round(LB[T]*YScale);

   for T := (ST+1) to ET do
   begin
      X2 := round((T-ST)*Xscale+30);
      YM2 := YZero-round(Mean[T]*YScale);
      YUB2 := YZero-round(UB[T]*YScale);
      YLB2 := YZero-round(LB[T]*YScale);
      SetColor(Green);
      Line(X1,YM1,X2,YM2);
      OutTextXY(X2-3,YM2-5,'.');
      SetColor(Red);
      Line(X1,YUB1,X2,YUB2);
      Line(X1,YLB1,X2,YLB2);
      OutTextXY(X2-3,YUB2-5,'.');
      OutTextXY(X2-3,YLB2-5,'.');
      X1 := X2;
      YM1 := YM2;
      YUB1 := YUB2;
      YLB1 := YLB2;
   end;
end;  {Procedure DrawCurve}

{++++++++++++++++++++++++++++++++++++++}

begin {Procedure GenGraph} {Verify Driver, Mode, and path}
   GraphDriver := VGA;     {adjust to the acceptable values}
```

```
   GraphMode    := VGAHi;
   InitGraph(GraphDriver,GraphMode,'a:\bgi\');
   SetBKColor(Black);
   ClearDevice;
   DrawAxis;
   YScale := 400/(Max-Min);
   XScale := 600/(ET-ST);
   YZero  := round(abs(Max)*YScale)+30;
   Line(30,YZero,630,YZero);
   SetLineStyle(DottedLn,0,NormWidth);
   SetYScale(YScale,YZero);
   SetXScale(XScale);

   SetLineStyle(SolidLn,0,NormWidth);
   DrawCurve(YScale,XScale);
   Line(600,25,630,25);
   SetColor(Green);
   Line(600,15,630,15);
   SetColor(White);
   OutTextXY(500,10,'Mean');
   OutTextXY(500,20,'Mean+(-)2SD');
   OutTextXY(400,470,'*** Press <Enter> to exit ***');
   readln;
   CloseGraph;
end;  {Procedure GenGraph}

{+++PART6 : TWO METHODS & MAIN PROGRAM+++++++++++++++++++++}
{   It includes:
    BuildDataArray,
    Likelihood,
    Bayesian,
    Main Program.                                          }

Procedure BuildDataArray;

{   This procedure build the array of number of failures
(maintenance actions)                                      }

Procedure WriteRealVec (K, H : integer;
                        XVec,YVec : RealVec;
                        var Output : text);
var T : integer;
    MF : real;
begin
   writeln(Output,'"Mean number of"   "Mean number of"  ');
   writeln(Output,'    Failures      Maintenance Actions');
   writeln(Output);
   for T := K to H do
      writeln(Output,XVec[T]:12:3,YVec[T]:18:3);
   writeln(Output);
   MF := NS*UPM;
```

```
   if Data = 1 then
      for T := H downto K do
         writeln(Output2,MF/XVec[T],MF/YVec[T]);
end;

{++++++++++++++++++++++++++++++++++++++}

begin {Procedure BuildDataArray}
   Max := 0;
   Min := 1E10;
   if Data = 0 then
      ReadData(Input,NOFVec,NOMAVec)
   else
   begin
      BuildVec1(ET,SetCF,SetSigmaSqrF,
                SetMuF,SetEtaF,NOFVec);
      BuildVec1(ET,SetCMA,SetSigmaSqrMA,
                SetMuMA,SetEtaMA,NOMAVec);
   end;
   GenDataGraph;
   WriteRealVec(1,ET,NOFVec,NOMAVec,Output);
end;  {Procedure BuildDataArray}

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure Title;

begin
   if Meth = 0 then
   begin
      writeln('Maximum likelihood procedure :');
      writeln(Output,'Maximum likelihood procedure :');
   end
   else
   begin
      writeln('Bayesian procedure :');
      writeln(Output,'Bayesian procedure :');
   end;

   writeln('The best time for subsystem upgrade');
   writeln('and assessing uncertainty');
   writeln('(from time ',ST,' to time ',ET,')');
   writeln;
   write  ('"Time"  "Best "  "  Cost  "  "Std.Dev."');
   writeln('  "Mean-2SD"  "Mean+2SD"');
   write  (' Index  upgrade   advantage    of cost ');
   writeln('    bound         bound    ');
   writeln('           time     of upgrade');

   writeln(Output,'The best time for subsystem upgrade');
   writeln(Output,'and assessing uncertainty');
```

```
   writeln(Output,'(from time ',ST,' to time ',ET,')');
   writeln(Output);
   write  (Output,'"Time"  "Best "  "  Cost  "  "Std.Dev."');
   writeln(Output,'  "Mean-2SD"  "Mean+2SD"');
   write  (Output,' Index  upgrade  advantage   of cost ');
   writeln(Output,'    bound        bound   ');
   writeln(Output,'            time    of upgrade');
end;

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   This procedure applies maximum likelihood model.      }

Procedure Likelihood;

begin
   Max := 0;
   Min := 0;
   Title;
   for Time := ST to ET do
   begin
      FindHatParameter1(Time,HatCF,NOFVec,MuF,EtaF,
                        SigmaSqrF,HatMuF,HatEtaF);
      FindHatParameter1(Time,HatCMA,NOMAVec,MuMA,EtaMA,
                        SigmaSqrMA,HatMuMA,HatEtaMA);
      FindChangeTime1(Hor,Time+LeadTime,HatCF,HatCMA,
                      HatMuF,HatEtaF,HatMuMA,HatEtaMA);

      Total := 0;
      ResetParameter1;
      if Rep > 0 then
         for R := 1 to Rep do
         begin
            BuildVec(Time,SetCF,SetSigmaSqrF,
                     SetMuF,SetEtaF,SimuFVec);
            BuildVec(Time,SetCMA,SetSigmaSqrMA,
                     SetMuMA,SetEtaMA,SimuMAVec);
            FindHatParameter1(Time,HatCF,SimuFVec,MuF,EtaF,
                              SigmaSqrF,HatMuF,HatEtaF);
            FindHatParameter1(Time,HatCMA,SimuMAVec,MuMA,
                              EtaMA,SigmaSqrMA,HatMuMA,
                              HatEtaMA);
            Uncertainty1(Hor,Time+LeadTime,R,HatCF,HatCMA,
                         HatMuF,HatEtaF,HatMuMA,HatEtaMA);
         end
      else
      begin
         writeln;
         writeln(Output);
      end;
   end;
   writeln(Output);
```

```
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   This procedure applies the Bayesian model.               }

Procedure Bayesian;

{++++++++++++++++++++++++++++++++++++}

Procedure Draw(var HatC : integer;
               var HatMu : real;
               var HatEta : real;
               Mu, Eta, RSqr, VSqr, RhoSqr, PiDist : RealVec);
var C : integer;
    U, Z, Cov : real;

begin
   U := Random;
   C := 1;
   while U > PiDist[C] do
      C := C+1;
   HatC := C;
   Z := GenNormal;
   HatMu := Mu[C]+(sqrt(RSqr[C])*Z);
   if C < Time then
   begin
      Cov := (-sqrt(RhoSqr[C])*Z)
            +(sqrt(1-RhoSqr[C])*GenNormal);
      HatEta := Eta[C]+(sqrt(VSqr[C])*Cov);
   end
   else
      HatEta := 0;
end;

{++++++++++++++++++++++++++++++++++++++}

begin
   Max := 0;
   Min := 0;
   Title;
   InitialSigmaSqr(NOFVec,MuF,EtaF,SigmaSqrF);
   InitialSigmaSqr(NOMAVec,MuMA,EtaMA,SigmaSqrMA);
   for Time := ST to ET do
   begin
      FindHatParameter2(Time,NOFVec,MuF,EtaF,SigmaSqrF,
                        VSqrF,RSqrF,RhoSqrF,PiStarF);
      FindHatParameter2(Time,NOMAVec,MuMA,EtaMA,SigmaSqrMA,
                        VSqrMA,RSqrMA,RhoSqrMA,PiStarMA);
      FindChangeTime2(Hor,Time+LeadTime,MuF,EtaF,
                      MuMA,EtaMA,PiStarF,PiStarMA);
```

```
      Total := 0;
      ResetParameter2;
      if Rep > 0 then
         Uncertainty2 (Hor,Time+LeadTime,MuF,EtaF,MuMA,EtaMA,
                       RSqrF,VSqrF,RhoSqrF,RSqrMA,VSqrMA,
                       RhoSqrMA,PiStarF,PiStarMA)
{         for R := 1 to Rep do
         begin
            Draw(HatCF,HatMuF,HatEtaF,MuF,EtaF,
                 RSqrF,VSqrF,RhoSqrF,PiDistF);
            Draw(HatCMA,HatMuMA,HatEtaMA,MuMA,EtaMA,
                 RSqrMA,VSqrMA,RhoSqrMA,PiDistMA);
            Uncertainty1(Hor,Time+LeadTime,R,HatCF,HatCMA,
                         HatMuF,HatEtaF,HatMuMA,HatEtaMA);
         end}      {You may use the procedure in the pair of }
      else         {braces instead of using procedure        }
      begin        {Uncertainty2. The procedure in the braces}
         writeln; {uses the resampling method to assess      }
         writeln(Output); {uncertainty.                      }
      end;         {Be sure to put a pair of braces on one of}
   end;            {the two procedures.                      }
   writeln(Output);
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

Procedure ViewGraph;
var Ch : char;

begin
   writeln;
   writeln('Do you want to see the graph');
   write('<y/n>? ');
   readln(Ch);
   if (Ch ='Y') or (Ch = 'y') then
      GenGraph;
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   This procedure gives the choise to reuse the simulated
data.                                                        }

Procedure Again;
var Ch : char;

begin
   writeln;
   writeln('Do you want to use the simulating data again');
   writeln('Data will be contained in "Simu.data" file');
   write('<y/n>? ');
   readln(Ch);
```

```
      if (Ch = 'Y') or (Ch = 'y') then
      begin
         reset(Output2);
         Data := 0;
         Infile := 'a:\Simu.Data'
      end;
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
{   This Function let user decide to terminate or continue the
job.                                                        }

Function Done : boolean;
var Ch : char;

begin
   repeat
      writeln;
      writeln('Enter 0 to stop or 1 to continue.');
      write('? ');
      readln(Ch);
   until (Ch = '0') or (Ch = '1');
   Done := (Ch = '0');
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

begin {Main Program}
   assign (Output,'a:\F.out');
   assign (Output2,'a:\Simu.Data');
   rewrite (Output);
   rewrite (Output2);

   DefaultParameter;
   SetUnitCost;
   SetMission;

   Randomize;
   RanSeed := 1; {User may change the value of RanSeed}

   repeat
      SetParameter;
      WriteParameter;
      BuildDataArray;
      if Meth = 0 then
         Likelihood
      else
         Bayesian;
      if Rep <> 0 then
         ViewGraph;
      if Data = 1 then
```

```
            Again;
      until Done;

      close(Output);
      close(Output2);
   end. {Main Program}
```

LIST OF REFERENCES

Cdr. C. Wrestler, Private Communication, December 1992.

Carlin, B. P., A. E. Gelfand, and A. F. M. Smith, Hierarchical Bayesian analysis of changepoint problems, Applied Statistics, 41, pp. 389-408, 1992.

G. S. Fishman, Principles of discrete event simulation, 1978.

Donald P. Gaver and Patricia A. Jacobs, Statistical approaches to detection and quantification of a trend with return-on-investment application, Naval Postgraduate School Technical Report NPSOR-93-007, December 1992.

Smith, A. F. M., A Bayesian Approach to inference about a change-point in a sequence of random variables, Biometrika, 62, pp. 407-416, 1975.

Efron, B. and R. Tibshirani, Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, Statistical Science, 1, pp. 54-77. 1986.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3.	Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5002	
	Attn: Chairman	1
	Attn: Prof. Donald P. Gaver	1
	Attn: Prof. Patricia A. Jacobs	1
	Attn: Prof. So Young Sohn	1
4.	Library Chung Cheng Institute of Technology Taoyuan, Taiwan, R.O.C.	2
5.	LCDR. Chang Tun-Jen Chung Cheng Institute of Technology Taoyuan, Taiwan, R.O.C.	2
6.	Capt. J. Gill NAVAIR (AIR 419) Washington, D.C. 20550	1
7.	C. Wrestler NAVAIR (AIR 419) Washington, D.C. 20550	1